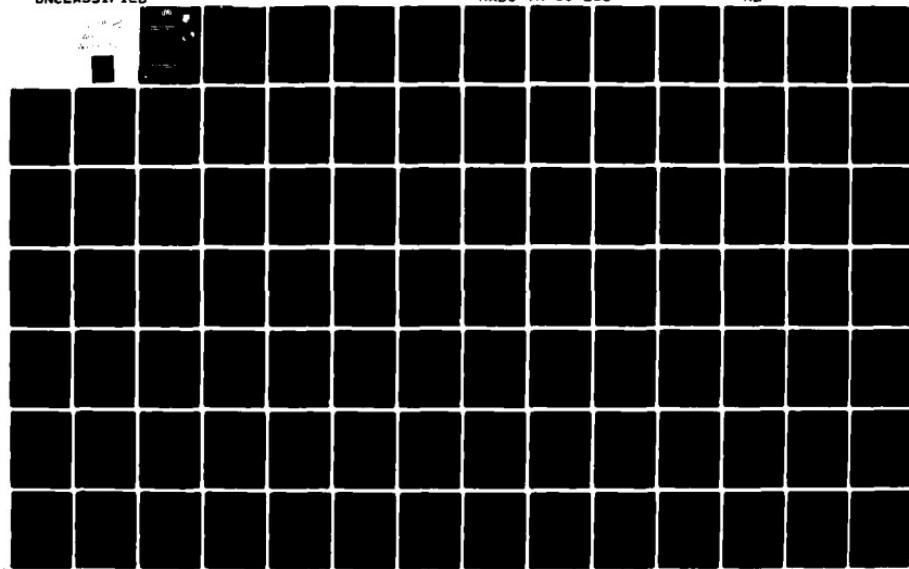


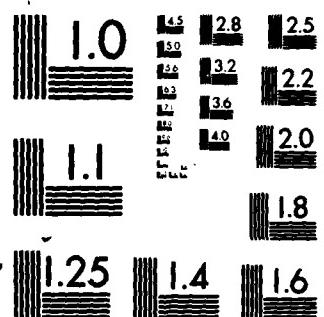
AD-A089 702 CLARKSON COLL OF TECHNOLOGY POTSDAM N Y DEPT OF ELEC--ETC F/G 17/9  
MODELING OF SHADOWS IN RADAR CLUTTER. (U)  
JUL 80 B A BLACK, M AROZULLAH, W LADEW F30602-78-C-0102

UNCLASSIFIED

RADC-TR-80-238

NL





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

AD A089702

REPORT OF SURVEY  
OF THE CHINESE  
PROVINCE OF  
SZECHUAN  
BY  
GENERAL  
JOHN C. WOODWARD,  
U.S.A.

1868

1869

1870

RECORDED IN THE  
LIBRARY OF THE  
UNIVERSITY OF TORONTO

1870

1871

1872

1873

1874

1875

1876

1877

1878

1879

1880

1881

1882

1883

1884

1885

1886

1887

1888

1889

1890

1891

1892

1893

1894

1895

1896

1897

1898

1899

1900

1901

1902

1903

1904

1905

1906

1907

1908

1909

1910

1911

1912

1913

1914

1915

1916

1917

1918

1919

1920

1921

1922

1923

1924

1925

1926

1927

1928

1929

1930

1931

1932

1933

1934

1935

1936

1937

1938

1939

1940

1941

1942

1943

1944

1945

1946

1947

1948

1949

1950

1951

1952

1953

1954

1955

1956

1957

1958

1959

1960

1961

1962

1963

1964

1965

1966

1967

1968

1969

1970

1971

1972

1973

1974

1975

1976

1977

1978

1979

1980

1981

1982

1983

1984

1985

1986

1987

1988

1989

1990

1991

1992

1993

1994

1995

1996

1997

1998

1999

2000

2001

2002

2003

2004

2005

2006

2007

2008

2009

2010

2011

2012

2013

2014

2015

2016

2017

2018

2019

2020

2021

2022

2023

2024

2025

2026

2027

2028

2029

2030

2031

2032

2033

2034

2035

2036

2037

2038

2039

2040

2041

2042

2043

2044

2045

2046

2047

2048

2049

2050

2051

2052

2053

2054

2055

2056

2057

2058

2059

2060

2061

2062

2063

2064

2065

2066

2067

2068

2069

2070

2071

2072

2073

2074

2075

2076

2077

2078

2079

2080

2081

2082

2083

2084

2085

2086

2087

2088

2089

2090

2091

2092

2093

2094

2095

2096

2097

2098

2099

2100

2101

2102

2103

2104

2105

2106

2107

2108

2109

2110

2111

2112

2113

2114

2115

2116

2117

2118

2119

2120

2121

2122

2123

2124

2125

2126

2127

2128

2129

2130

2131

2132

2133

2134

2135

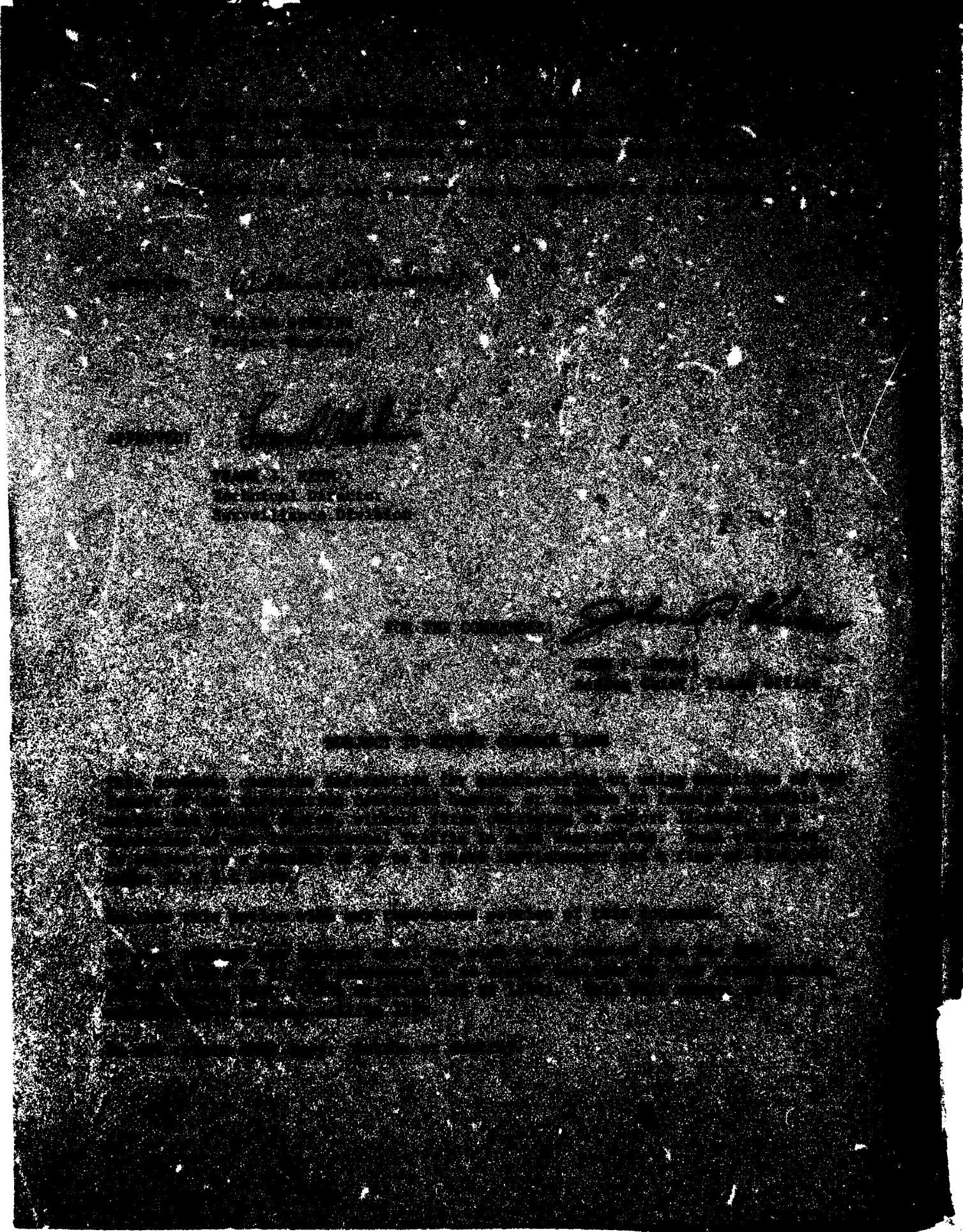
2136

2137

2138

2139

</



## UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(1) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC TR-80-238	2. GOVT ACCESSION NO. AD-A089	3. RECIPIENT'S CATALOG NUMBER 702
4. TITLE (Title Subtitle) MODELING OF SHADOWS IN RADAR CLUTTER		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 1 Jun 79 - 31 Dec 79
6. PERFORMING ORG. REPORT NUMBER N/A		7. CONTRACT OR GRANT NUMBER(S) F30602-78-C-0102
8. AUTHOR(s) Bruce A. Black Mohammed Arozullah William Ladew		9. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 450611PE
10. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (OCTS) Griffiss AFB NY 13441		11. REPORT DATE July 180
12. MONITORING AGENCY NAME & ADDRESS/If different from Controlling Office) Same		13. NUMBER OF PAGES 114
14. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
16. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, If different from Report) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: William L. Simkins (OCTS)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Clutter Ground Clutter Shadowing (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (21) (22) (23) (24) (25) (26) (27) (28) (29) (30) (31) (32) (33) (34) (35) (36) (37) (38) (39) (40) (41) (42) (43) (44) (45) (46) (47) (48) (49) (50) (51) (52) (53) (54) (55) (56) (57) (58) (59) (60) (61) (62) (63) (64) (65) (66) (67) (68) (69) (70) (71) (72) (73) (74) (75) (76) (77) (78) (79) (80) (81) (82) (83) (84) (85) (86) (87) (88) (89) (90) (91) (92) (93) (94) (95) (96) (97) (98) (99) (100) (101) (102) (103) (104) (105) (106) (107) (108) (109) (110) (111) (112) (113) (114) (115) (116) (117) (118) (119) (120) (121) (122) (123) (124) (125) (126) (127) (128) (129) (130) (131) (132) (133) (134) (135) (136) (137) (138) (139) (140) (141) (142) (143) (144) (145) (146) (147) (148) (149) (150) (151) (152) (153) (154) (155) (156) (157) (158) (159) (160) (161) (162) (163) (164) (165) (166) (167) (168) (169) (170) (171) (172) (173) (174) (175) (176) (177) (178) (179) (180) (181) (182) (183) (184) (185) (186) (187) (188) (189) (190) (191) (192) (193) (194) (195) (196) (197) (198) (199) (200) (201) (202) (203) (204) (205) (206) (207) (208) (209) (210) (211) (212) (213) (214) (215) (216) (217) (218) (219) (220) (221) (222) (223) (224) (225) (226) (227) (228) (229) (230) (231) (232) (233) (234) (235) (236) (237) (238) (239) (240) (241) (242) (243) (244) (245) (246) (247) (248) (249) (250) (251) (252) (253) (254) (255) (256) (257) (258) (259) (260) (261) (262) (263) (264) (265) (266) (267) (268) (269) (270) (271) (272) (273) (274) (275) (276) (277) (278) (279) (280) (281) (282) (283) (284) (285) (286) (287) (288) (289) (290) (291) (292) (293) (294) (295) (296) (297) (298) (299) (300) (301) (302) (303) (304) (305) (306) (307) (308) (309) (310) (311) (312) (313) (314) (315) (316) (317) (318) (319) (320) (321) (322) (323) (324) (325) (326) (327) (328) (329) (330) (331) (332) (333) (334) (335) (336) (337) (338) (339) (340) (341) (342) (343) (344) (345) (346) (347) (348) (349) (350) (351) (352) (353) (354) (355) (356) (357) (358) (359) (360) (361) (362) (363) (364) (365) (366) (367) (368) (369) (370) (371) (372) (373) (374) (375) (376) (377) (378) (379) (380) (381) (382) (383) (384) (385) (386) (387) (388) (389) (390) (391) (392) (393) (394) (395) (396) (397) (398) (399) (400) (401) (402) (403) (404) (405) (406) (407) (408) (409) (410) (411) (412) (413) (414) (415) (416) (417) (418) (419) (420) (421) (422) (423) (424) (425) (426) (427) (428) (429) (430) (431) (432) (433) (434) (435) (436) (437) (438) (439) (440) (441) (442) (443) (444) (445) (446) (447) (448) (449) (450) (451) (452) (453) (454) (455) (456) (457) (458) (459) (460) (461) (462) (463) (464) (465) (466) (467) (468) (469) (470) (471) (472) (473) (474) (475) (476) (477) (478) (479) (480) (481) (482) (483) (484) (485) (486) (487) (488) (489) (490) (491) (492) (493) (494) (495) (496) (497) (498) (499) (500) (501) (502) (503) (504) (505) (506) (507) (508) (509) (510) (511) (512) (513) (514) (515) (516) (517) (518) (519) (520) (521) (522) (523) (524) (525) (526) (527) (528) (529) (530) (531) (532) (533) (534) (535) (536) (537) (538) (539) (540) (541) (542) (543) (544) (545) (546) (547) (548) (549) (540) (541) (542) (543) (544) (545) (546) (547) (548) (549) (550) (551) (552) (553) (554) (555) (556) (557) (558) (559) (550) (551) (552) (553) (554) (555) (556) (557) (558) (559) (560) (561) (562) (563) (564) (565) (566) (567) (568) (569) (560) (561) (562) (563) (564) (565) (566) (567) (568) (569) (570) (571) (572) (573) (574) (575) (576) (577) (578) (579) (570) (571) (572) (573) (574) (575) (576) (577) (578) (579) (580) (581) (582) (583) (584) (585) (586) (587) (588) (589) (580) (581) (582) (583) (584) (585) (586) (587) (588) (589) (590) (591) (592) (593) (594) (595) (596) (597) (598) (599) (590) (591) (592) (593) (594) (595) (596) (597) (598) (599) (600) (601) (602) (603) (604) (605) (606) (607) (608) (609) (600) (601) (602) (603) (604) (605) (606) (607) (608) (609) (610) (611) (612) (613) (614) (615) (616) (617) (618) (619) (610) (611) (612) (613) (614) (615) (616) (617) (618) (619) (620) (621) (622) (623) (624) (625) (626) (627) (628) (629) (620) (621) (622) (623) (624) (625) (626) (627) (628) (629) (630) (631) (632) (633) (634) (635) (636) (637) (638) (639) (630) (631) (632) (633) (634) (635) (636) (637) (638) (639) (640) (641) (642) (643) (644) (645) (646) (647) (648) (649) (640) (641) (642) (643) (644) (645) (646) (647) (648) (649) (650) (651) (652) (653) (654) (655) (656) (657) (658) (659) (650) (651) (652) (653) (654) (655) (656) (657) (658) (659) (660) (661) (662) (663) (664) (665) (666) (667) (668) (669) (660) (661) (662) (663) (664) (665) (666) (667) (668) (669) (670) (671) (672) (673) (674) (675) (676) (677) (678) (679) (670) (671) (672) (673) (674) (675) (676) (677) (678) (679) (680) (681) (682) (683) (684) (685) (686) (687) (688) (689) (680) (681) (682) (683) (684) (685) (686) (687) (688) (689) (690) (691) (692) (693) (694) (695) (696) (697) (698) (699) (690) (691) (692) (693) (694) (695) (696) (697) (698) (699) (700) (701) (702) (703) (704) (705) (706) (707) (708) (709) (700) (701) (702) (703) (704) (705) (706) (707) (708) (709) (710) (711) (712) (713) (714) (715) (716) (717) (718) (719) (710) (711) (712) (713) (714) (715) (716) (717) (718) (719) (720) (721) (722) (723) (724) (725) (726) (727) (728) (729) (720) (721) (722) (723) (724) (725) (726) (727) (728) (729) (730) (731) (732) (733) (734) (735) (736) (737) (738) (739) (730) (731) (732) (733) (734) (735) (736) (737) (738) (739) (740) (741) (742) (743) (744) (745) (746) (747) (748) (749) (740) (741) (742) (743) (744) (745) (746) (747) (748) (749) (750) (751) (752) (753) (754) (755) (756) (757) (758) (759) (750) (751) (752) (753) (754) (755) (756) (757) (758) (759) (760) (761) (762) (763) (764) (765) (766) (767) (768) (769) (760) (761) (762) (763) (764) (765) (766) (767) (768) (769) (770) (771) (772) (773) (774) (775) (776) (777) (778) (779) (770) (771) (772) (773) (774) (775) (776) (777) (778) (779) (780) (781) (782) (783) (784) (785) (786) (787) (788) (789) (780) (781) (782) (783) (784) (785) (786) (787) (788) (789) (790) (791) (792) (793) (794) (795) (796) (797) (798) (799) (790) (791) (792) (793) (794) (795) (796) (797) (798) (799) (800) (801) (802) (803) (804) (805) (806) (807) (808) (809) (800) (801) (802) (803) (804) (805) (806) (807) (808) (809) (810) (811) (812) (813) (814) (815) (816) (817) (818) (819) (810) (811) (812) (813) (814) (815) (816) (817) (818) (819) (820) (821) (822) (823) (824) (825) (826) (827) (828) (829) (820) (821) (822) (823) (824) (825) (826) (827) (828) (829) (830) (831) (832) (833) (834) (835) (836) (837) (838) (839) (830) (831) (832) (833) (834) (835) (836) (837) (838) (839) (840) (841) (842) (843) (844) (845) (846) (847) (848) (849) (840) (841) (842) (843) (844) (845) (846) (847) (848) (849) (850) (851) (852) (853) (854) (855) (856) (857) (858) (859) (850) (851) (852) (853) (854) (855) (856) (857) (858) (859) (860) (861) (862) (863) (864) (865) (866) (867) (868) (869) (860) (861) (862) (863) (864) (865) (866) (867) (868) (869) (870) (871) (872) (873) (874) (875) (876) (877) (878) (879) (870) (871) (872) (873) (874) (875) (876) (877) (878) (879) (880) (881) (882) (883) (884) (885) (886) (887) (888) (889) (880) (881) (882) (883) (884) (885) (886) (887) (888) (889) (890) (891) (892) (893) (894) (895) (896) (897) (898) (899) (890) (891) (892) (893) (894) (895) (896) (897) (898) (899) (900) (901) (902) (903) (904) (905) (906) (907) (908) (909) (900) (901) (902) (903) (904) (905) (906) (907) (908) (909) (910) (911) (912) (913) (914) (915) (916) (917) (918) (919) (910) (911) (912) (913) (914) (915) (916) (917) (918) (919) (920) (921) (922) (923) (924) (925) (926) (927) (928) (929) (920) (921) (922) (923) (924) (925) (926) (927) (928) (929) (930) (931) (932) (933) (934) (935) (936) (937) (938) (939) (930) (931) (932) (933) (934) (935) (936) (937) (938) (939) (940) (941) (942) (943) (944) (945) (946) (947) (948) (949) (940) (941) (942) (943) (944) (945) (946) (947) (948) (949) (950) (951) (952) (953) (954) (955) (956) (957) (958) (959) (950) (951) (952) (953) (954) (955) (956) (957) (958) (959) (960) (961) (962) (963) (964) (965) (966) (967) (968) (969) (960) (961) (962) (963) (964) (965) (966) (967) (968) (969) (970) (971) (972) (973) (974) (975) (976) (977) (978) (979) (970) (971) (972) (973) (974) (975) (976) (977) (978) (979) (980) (981) (982) (983) (984) (985) (986) (987) (988) (989) (980) (981) (982) (983) (984) (985) (986) (987) (988) (989) (990) (991) (992) (993) (994) (995) (996) (997) (998) (999) (990) (991) (992) (993) (994) (995) (996) (997) (998) (999) (1000) (1001) (1002) (1003) (1004) (1005) (1006) (1007) (1008) (1009) (1000) (1001) (1002) (1003) (1004) (1005) (1006) (1007) (1008) (1009) (1010) (1011) (1012) (1013) (1014) (1015) (1016) (1017) (1018) (1019) (1010) (1011) (1012) (1013) (1014) (1015) (1016) (1017) (1018) (1019) (1020) (1021) (1022) (1023) (1024) (1025) (1026) (1027) (1028) (1029) (1020) (1021) (1022) (1023) (1024) (1025) (1026) (1027) (1028) (1029) (1030) (1031) (1032) (1033) (1034) (1035) (1036) (1037) (1038) (1039) (1030) (1031) (1032) (1033) (1034) (1035) (1036) (1037) (1038) (1039) (1040) (1041) (1042) (1043) (1044) (1045) (1046) (1047) (1048) (1049) (1040) (1041) (1042) (1043) (1044) (1045) (1046) (1047) (1048) (1049) (1050) (1051) (1052) (1053) (1054) (1055) (1056) (1057) (1058) (1059) (1050) (1051) (1052) (1053) (1054) (1055) (1056) (1057) (1058) (1059) (1060) (1061) (1062) (1063) (1064) (1065) (1066) (1067) (1068) (1069) (1060) (1061) (1062) (1063) (1064) (1065) (1066) (1067) (1068) (1069) (1070) (1071) (1072) (1073) (1074) (1075) (1076) (1077) (1078) (1079) (1070) (1071) (1072) (1073) (1074) (1075) (1076) (1077) (1078) (1079) (1080) (1081) (1082) (1083) (1084) (1085) (1086) (1087) (1088) (1089) (1080) (1081) (1082) (1083) (1084) (1085) (1086) (1087) (1088) (1089) (1090) (1091) (1092) (1093) (1094) (1095) (1096) (1097) (1098) (1099) (1090) (1091) (1092) (1093) (1094) (1095) (1096) (1097) (1098) (1099) (1100) (1101) (1102) (1103) (1104) (1105) (1106) (1107) (1108) (1109) (1100) (1101) (1102) (1103) (1104) (1105) (1106) (1107) (1108) (1109) (1110) (1111) (1112) (1113) (1114) (1115) (1116) (1117) (1118) (1119) (1110) (1111) (1112) (1113) (1114) (1115) (1116) (1117) (1118) (1119) (1120) (1121) (1122) (1123) (1124) (1125) (1126) (1127) (1128) (1129) (1120) (1121) (1122) (1123) (1124) (1125) (1126) (1127) (1128) (1129) (1130) (1131) (1132) (1133) (1134) (1135) (1136) (1137) (1138) (1139) (1130) (1131) (1132) (1133) (1134) (1135) (1136) (1137) (1138) (1139) (1140) (1141) (1142) (1143) (1144) (1145) (1146) (1147) (1148) (1149) (1140) (1141) (1142) (1143) (1144) (1145) (1146) (1147) (1148) (1149) (1150) (1151) (1152) (1153) (1154) (1155) (1156) (1157) (1158) (1159) (1150) (1151) (1152) (1153) (1154) (1155) (1156) (1157) (1158) (1159) (1160) (1161) (1162) (1163) (1164) (1165) (1166) (1167) (1168) (1169) (1160) (1161) (1162) (1163) (1164) (1165) (1166) (1167) (1168) (1169) (1170) (1171) (1172) (1173) (1174) (1175) (1176) (1177) (1178) (1179) (1170) (1171) (1172) (1173) (1174) (1175) (1176) (1177) (1178) (1179) (1180) (1181) (1182) (1183) (1184) (1185) (1186) (1187) (1188) (1189) (1180) (1181) (1182) (1183) (1184) (1185) (1186) (1187) (1188) (1189) (1190) (1191) (1192) (1193) (1194) (1195) (1196) (1197) (1198) (1199) (1190) (1191) (1192) (1193) (1194) (1195) (1196) (1197) (1198) (1199) (1200) (1201) (1202) (1203) (1204) (1205) (1206) (1207) (1208) (1209) (1200) (1201) (1202) (1203) (1204) (1205) (1206) (1207) (1208) (1209) (1210) (1211) (1212) (1213) (1214) (1215) (1216) (1217) (1218) (1219) (1210) (1211) (1212) (1213) (1214) (1215) (1216) (1217) (1218) (1219) (1220) (1221) (1222) (1223) (1224) (1225) (1226) (1227) (1228) (1229) (1220) (1221) (1222) (1223) (1224) (1225) (1226) (1227) (1228) (1229) (1230) (1231) (1232) (1233) (1234) (1235) (1236) (1237) (1238) (1239) (1230) (1231) (1232) (1233) (1234) (1235) (1236) (1237) (1238) (1239) (1240) (1241) (1242) (1243) (1244) (1245) (1246) (1247) (1248) (1249) (1240) (1241) (1242) (1243) (1244) (1245) (1246) (1247) (1248) (1249) (1250) (1251) (1252) (1253) (1254) (1255) (1256) (1257) (1258) (1259) (1250) (1251) (1252) (1253) (1254) (1255) (1256) (1257) (1258) (1259) (1260) (1261) (1262) (1263) (1264) (1265) (1266) (1267) (1268) (1269) (1260) (1261) (1262) (1263) (1264) (1265) (1266) (1267) (1268) (1269) (1270) (1271) (1272) (1273) (1274) (1275) (1276) (1277) (1278) (1279) (1270) (1271) (1272) (1273) (1274) (1275) (1276) (1277) (1278) (1279) (1280) (1281) (1282) (1283) (1284)		

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

**Item 20 (Cont'd)**

Laboratory at RADC. The algorithms were run on the IBM 360/65 at Clarkson College using programs either written at Clarkson or obtained from the pattern recognition package ARTHUR.

The Forgy-Jancey algorithm was found to be flexible and capable of handling large data sets at moderate cost, but was inconsistent in its ability to detect the "natural" clusters in unknown data. The minimum spanning tree algorithm was more reliable in this regard, but the ARTHUR implementation was found to be costly in terms of required storage. The report contains numerous figures illustrating the application of the two clustering algorithms to various configurations of data.

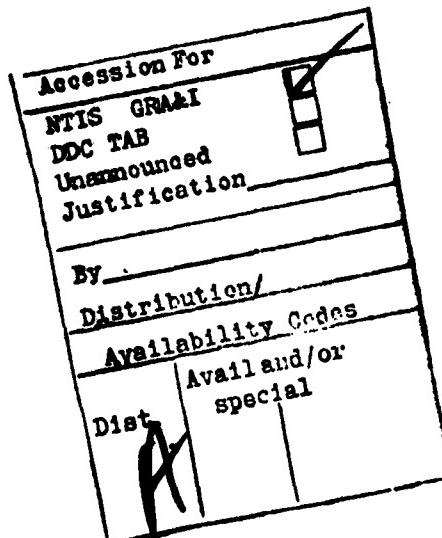
Clustering techniques appear to hold promise for examining the structure of unknown clutter data, but the results must be interpreted with care, since a natural clustering is not always found. The primary result of the work to date has been the installation of the programs on the Clarkson College computer and the familiarization of the researchers with their use and limitations. It now remains to apply these programs to a wider class of data.

**UNCLASSIFIED**

**SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)**

## CONTENTS

1.0 Introduction	page	1
2.0 The Forgy-Jancey Algorithm	6	
2.1 The Algorithm	6	
2.2 Testing	10	
2.3 Removal of Outliers	16	
2.4 Testing of Actual Radar Data	18	
2.5 Conclusions	21	
3.0 Minimal Spanning Tree Algorithm	23	
3.1 Introduction	23	
3.2 Minimal Spanning Trees	24	
3.3 ARTHUR and the MST	26	
3.4 Actual Clustering Using TREE	28	
3.5 Conclusions	30	
4.0 Preprocessing Techniques	32	
4.1 Introduction	32	
4.2 Point-to-Point Local Variance	33	
4.3 Point-to-Point Local Gradient	35	
4.4 Radial Derivative Method	37	
4.5 Conclusions	42	
4.6 Note	43	
5.0 Conclusions	44	
Appendix: Program Listings	47	
Figures	49	
References	105	



## MODELING OF SHADOWS IN RADAR CLUTTER

### LIST OF FIGURES

Figure	Description	page
2.1:	The Forgy-Jancey Algorithm	49
2.2:	Forgy-Jancey Program Path	50
2.3:	Test Scene for Clustering	51
2.4:	Intensity Clustering of Test Scene	52
2.5:	x-y Clustering of Test Scene	53
2.6:	Intensity Clustering of Simulated Shadows	54
2.7:	Intensity Clustering; Intensity in dB	55
2.8:	x-y Clustering of Simulated Shadows	56
2.9:	Intensity Clustering; Low Background Level	57
2.10:	Low Background Level Run to Convergence	58
2.11:	x-y Clustering, Low Background Level; NC=2	59
2.12:	x-y Clustering, Low Background Level; NC=3	60
2.13:	x-y Clustering, Low Background Level; NC=4	61
2.14:	x-y Clustering, Low Background Level; NC=5	62
2.15:	x-y Clustering, Low Background Level; NC=6	63
2.16:	x-y Clustering, Low Background Level; NC=7	64
2.17:	Total Summed Deviations	65
2.18:	Test Scene With Outliers	66
2.19:	Radar Data	67
2.20:	Low Intensity Regions	68
2.21:	x-y Clustering of Low Intensity Regions	69
3.1:	TREE Example Run	70
3.2:	Default Clustering	71
3.3:	Better Clustering	72
3.4:	Best Clustering Found	73
3.5:	Data Set for Tree Example	74
3.6:	Minimal Spanning Tree	75
3.7:	Best Clustering Found	76
4.1:	Point-to-Point Local Variance	77
4.2:	Point-to-Point Local Gradient	78
4.3:	Location of Radar in Upper Quadrant	79
4.4:	Location of Radar in Lower Quadrant	80
4.5:	Radial Derivative, Upper Quadrant	81
4.6:	Radial Derivative, Lower Quadrant	82
4.7:	Radial Derivative and Shadows	83
4.8:	Precursors of Shadows	84
A.1:	Forgy-Jancey Comments	85
A.2:	Outlier Comments	86
A.3:	Subroutine EXEC	87
A.4:	Subroutine EXEC2	88
A.5:	Subroutine EXEC3	89

## MODELING OF SHADOWS IN RADAR CLUTTER

### LIST OF FIGURES, CONTINUED

	page
Figure A.6 (Part 1): Subroutine KMEAN	90
Figure A.6 (Part 2): Subroutine KMEAN	91
Figure A.6 (Part 3): Subroutine KMEAN	92
Figure A.7 (Part 1): Subroutine KMEAN2	93
Figure A.7 (Part 2): Subroutine KMEAN2	94
Figure A.7 (Part 3): Subroutine KMEAN2	95
Figure A.8: Subroutine RESULT	96
Figure A.9: Subroutine RSULT2	97
Figure A.10: MAIN; USER; USEROUT; Function DIST	98
Figure A.11: MAIN; Two Versions of DIST	99
Figure A.12: Subroutine RAY	100
Figure A.13: Creates Rayleigh Data, SIGMA=10,000.	101
Figure A.14: Creates Rayleigh Data, SIGMA=50.	102
Figure A.15: Cluster Plot Program	103
FIGURE A.16: Grey Scale Program	104

## EVALUATION

1. As future tactical and strategic sensors become more automated and unattended, their processing will increasingly treat the environmental scatter as information rather than "clutter" or interference. By inference from measureable quantities and statistics, the processor will recognize the existence of weather, chaff, discrete targets, homogeneous distributed areas, shadowing as opposed to specular reflection, and other environmental categories. This information will allow the system to adapt its waveform, energy budget, and detection/CFAR algorithms for optimum performance. This effort investigated algorithms which may be usable in such a recognition scheme.
2. Stress was placed on the strengths, weaknesses, and peculiarities of two promising algorithms when applied to the recognition of shadowed areas. The program has provided an excellent insight into the robustness and sensitivities to data format of each algorithm and the care required in the interpretation and use of algorithmic results. This report will be a useful reference in the development of clutter model and associated adaptive processing for future intelligent unattended systems.

*William L. Simkins*

WILLIAM L. SIMKINS

Project Engineer

## MODELING OF SHADOWS IN RADAR CLUTTER

### 1.0 INTRODUCTION

This report presents the results of the work done during the period June-December 1979 on investigating the applicability of nonparametric clustering techniques to the extraction of features from radar clutter. The Forgy-Jancey [1] and minimum spanning tree [2] algorithms were investigated, with particular application to the problem of locating and characterizing shadowed regions in ground clutter. The significant result of the work is an exposition of the strengths and weaknesses of the two clustering techniques when applied to both simulated and actual radar returns. It is expected that the same strengths and weaknesses should appear in applications involving extraction of clutter features other than shadows.

The motivation for undertaking the present study of ground clutter is summarized in [3]. The need has existed since before the Second World War to extend the ability of radars to detect and identify targets of low cross-section in severe clutter. Initial efforts were concerned with increasing transmitter power, but for the past two decades the emphasis has been on development of the signal processing abilities of radar receivers [4]. The recent explosive developments in digital technology are expected to result in a new generation

## MODELING OF SHADOWS IN RADAR CLUTTER

of sophisticated signal-processing radars, able to take advantage of random-access memories and high-speed computational ability. The need to specify how such radars are to be designed and used provides a requirement for an improved understanding of the structure of backscatter clutter.

There are several indications that the present understanding of backscatter clutter is inadequate. First, clutter measurement efforts concerned with various terrain and cultural features have produced data of only limited consistency. In particular, it is not uncommon to find a 15 to 20 dB spread in measurements of intensity returns made at different times from supposedly identical terrain. Second, there has been only limited success in predicting backscatter from topographical maps, even though the terrain relief, surface features, and ground cover are specified on the maps. Finally, radars employing signal processing systems, such as CFAR, that are based on particular clutter models [5] have frequently been found to perform more poorly than analysis based on the clutter models would suggest.

There are several possible reasons for the poor consistency of measurement data and the difficulty of predicting backscatter intensity from terrain descriptions. First, it is likely that backscatter is highly sensitive to surface parameters such as moisture, temperature, and wind

## MODELING OF SHADOWS IN RADAR CLUTTER

velocity, that are difficult to measure simultaneously with the radar return, and have not always been included as a part of the recorded "ground truth". Second is the possibility that the clutter returns vary with the parameters of the radar itself in ways that have not yet been clearly recognized.

Much of the radar signal processing effort to date has been based on the assumption that clutter intensity can be modeled by a random variable. The motivation to develop such a model is strong, because of the success of design methods using the Neyman-Pearson decision rule in connection with a gaussian or rayleigh model for thermal noise. Such decision-theoretic methods could be easily extended to detection of targets in clutter, if only the probability density function of the clutter intensity could be found. It is likely, however, that ground clutter is too heterogeneous to be represented by a single random variable, or possibly even by a single type of random variable. It is therefore necessary that an examination of the structure of radar clutter be carried out in a way that does not rely on knowledge of a probability distribution.

Methods for investigating the structure of unknown data have been developed in connection with research in taxonomy [1,6] and chemistry [7]. These methods are referred to as either classification or clustering methods, depending on whether or not a statistical structure is assumed for the unknown data. The classification methods assume that each data

## MODELING OF SHADOWS IN RADAR CLUTTER

entity belongs to one of several possible categories, each category characterized by a probability density function. Most classification algorithms require training on data whose classification by category is known. The training allows the algorithm to estimate the density functions corresponding to the various categories. The estimated density functions are then used in classifying future unknown data. In contrast, clustering methods do not assume that categories of data can be characterized by density functions. Instead, data entities are grouped into clusters by examining the distances between entities based on some suitable measure of distance. A detailed discussion and comparison of many available classification and clustering methods is presented in [6].

In the present research the view has been taken that ground clutter from arbitrary sources can not be reliably represented in terms of probability density functions [8]. Attention was therefore focussed on an investigation of the various clustering methods. The research has proceeded along three paths: first an investigation of the Forgy-Jancey clustering algorithm, second an investigation of the minimum spanning tree algorithm contained in the program package ARTHUR, and, finally, an investigation into some pre-processing techniques applicable to either of the clustering algorithms.

## MODELING OF SHADOWS IN RADAR CLUTTER

A description of the Forgy-Jancey algorithm along with the results obtained from applying it to a variety of simulated and actual clutter returns is presented in Section 2. Although the Forgy-Jancey algorithm is well-known, it was necessary to create our own programs, and to devise from scratch the necessary input-output and display routines. Investigation of the minimum spanning tree clustering method is described in detail in Section 3. This investigation began with acquisition of the ARTHUR program package [7] from the University of Georgia. The necessary job-control language was created, and the ARTHUR package compiled so that it is currently available on the Clarkson College computer. The minimum spanning tree algorithm was tested on some of the same simulated data as was devised for the Forgy-Jancey tests. Section 4 contains the results of some pre-processing efforts that were tried on actual clutter data. The pre-processing was motivated by the need to reduce the quantity of data to a managable amount for the ARTHUR package. The pre-processing is typical of the steps that would be taken in an actual clustering effort, and provided an indication of the amounts of CPU time, programming time, and costs that are involved in such steps. Section 5 contains an overall summary and the conclusions.

Sections 1, 2, and 5 were written by Bruce A. Black, Section 3 by William Ladew, and Section 4 by Mohammed Arozullah.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 2.0 THE FORGY-JANCEY ALGORITHM

#### 2.1 The Algorithm

This section discusses the operation of the Forgy-Jancey clustering algorithm, and the tests that were made on simulated and actual radar data to probe the strengths and weaknesses of the algorithm as a tool for identifying the structure of ground clutter from shadowed terrain. The computer program for the Forgy-Jancey algorithm was obtained from Fordon [9]. Several modifications to the algorithm were made to allow greater flexibility in manipulating, saving, and displaying the input and output data. The algorithm employed in all of the tests is due to Forgy [10]. Jancey's version [1] contains a heuristic to avoid local minima, but was not needed in the present application. The basic Forgy-Jancey program along with the modifications is presented in block-diagram form below. A listing of the important subroutines is presented in an Appendix.

Figure 2.1 is a flowchart showing the operation of the Forgy-Jancey algorithm. This algorithm forms clusters of data entities, where each data entity is a vector of NV components. The number NV of components in a data entity (number of variables), the number NE of data entities, the number NC of

## MODELING OF SHADOWS IN RADAR CLUTTER

clusters, and the coordinates of the cluster centroids must all be specified by the user. In this version of the algorithm the number NC of clusters cannot be modified by the program; the program can neither create nor delete clusters. In operation, the program will partition the NE data entities into NC clusters. A measure of the compactness of the clusters that is used as a figure of merit by the program is the total summed deviations. For a given partition of the data into clusters, this is the sum over all the entities of the distance between each entity and the centroid of the cluster to which it is assigned.

Forgy's algorithm begins with an initial set of NC cluster centroids. In the main loop of the program, each data entity is assigned in turn to the nearest cluster centroid. When all of the entities have been assigned, the cluster centroids are recomputed. Each iteration requires a complete pass through the data. It is easily shown that a pass through the main loop cannot increase the value of the total summed deviations, so that the algorithm must converge to at least a local minimum. In each pass through the main loop the program counts the number of entities whose cluster assignment is changed. When the number of changes falls below a user-supplied parameter MINREL, the program halts. Except where otherwise noted, programs were run with MINREL equal to zero, corresponding to complete convergence.

## MODELING OF SHADOWS IN RADAR CLUTTER

The distances between entities and cluster centroids required by the algorithm are computed by a user-supplied subroutine DIST. Because this routine is externally supplied, the distance function can be configured to meet the specific application. In particular, it is possible to base the distance calculation on a subset of the components of the data entities. Both euclidean and non-euclidean distance measures are allowed.

The program path for the Forgy-Jancey algorithm is shown in Figure 2.2. The program MAIN and the subroutines EXEC, USER, RESULT, KMEAN, and USROUT were all supplied or modified as a part of the present effort. The program supplies as outputs:

1. A raw membership list, listing cluster membership for each data entity.
2. A sorted membership list, giving for each cluster the sequence numbers of its members.
3. A sorted data file. The data entities themselves are sorted into groupings by cluster membership. This file can be saved on disk, thus allowing the entities forming a single cluster to be isolated for further processing.
4. The program prints the final centroid locations for each cluster, the number of data entities belonging to each cluster, and the value of the total summed

## MODELING OF SHADOWS IN RADAR CLUTTER

deviations.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 2.2 Testing

For initial testing of the Forgy-Jancey algorithm, a set of 10,000 data entities representing a 100-point by 100-point "scene" was generated. Each data entity was a three-vector,  $(x, y, I)$ , where  $x$  and  $y$  represent the position of the entity in rectangular coordinates and  $I$  represents the intensity of the radar return from position  $(x, y)$ . Two rectangular regions were set aside in the 100x100 array to represent "shadows", one region occupying 651 points and the other occupying 121 points. The test scene is depicted in Figure 2.3.

For the initial test of the algorithm, the value 0 was given to the intensity component of data entities in the shadowed areas, while the value 1 was given to the intensity component of data entities in the unshadowed area. The algorithm was applied with the number of clusters NC set equal to two. The distance function DIST was set to compute the distance between two entities as the magnitude of the difference between their intensities. The output of the program is shown in Figure 2.4. The points marked 1 correspond to cluster 1, while the points marked with a dot correspond to cluster 2. Note that cluster 1 contains exactly the entities in the shadowed array.

A second clustering was carried out using only the entities in cluster 1 as data. This time the function DIST was

## MODELING OF SHADOWS IN RADAR CLUTTER

changed to compute the distance between two entities as the euclidean distance between their positions. The program was run with  $NC = 2$ . The output is shown in Figure 2.5. In this figure the points marked with a dot were not included in the input data. The points marked 1 and 2 represent the two clusters obtained by the algorithm. Note that these correspond exactly with the two shadows.

A second, more demanding, test of the Forgy-Jancey algorithm was made using random numbers as simulated intensity values. A program was written to produce a sequence of numbers having a rayleigh distribution. The distribution parameter sigma is entered as an input to the program. To establish the test data, 10,000 rayleigh-distributed numbers were generated with  $\sigma = 1$ . These numbers were assigned as intensity values to the data entities in the 100 by 100 array. Next a sequence of rayleigh-distributed numbers was generated with sigma much greater than one. These numbers were added to the intensity values of all the data entities outside the two shadowed areas. Thus a set of test data was established to represent two shadows containing only rayleigh-distributed thermal noise in a larger region of brighter rayleigh-distributed clutter.

As in the previous test, the Forgy-Jancey algorithm was applied twice, once to separate the low-intensity shadows from the bright clutter background, and a second time to separate

## MODELING OF SHADOWS IN RADAR CLUTTER

the low-intensity cluster into individual shadows. The result of the first clustering is shown in Figure 2.6. The bright clutter in this case has sigma set to 10,000. In the figure, the dark cluster is indicated by ones and the bright cluster by twos. Note that the dark cluster includes many points belonging to the bright region. In fact, the algorithm was unable in this case to locate the dark region; the program was instructed to find two clusters, and it did so by splitting the bright data half. The shadow data are simply included in the darker of the two clusters.

To provide a greater apparent separation between the dark and bright data, and to provide an example of the use of coordinate scaling, the program was modified to interpret intensity values in decibels. This was effected by employing the program USER to replace the intensity values  $I$  for each entity by their logarithms  $\ln I$  as the entities were read into core. The result of a clustering done on the rayleigh data using this logarithmic transformation is shown in Figure 2.7. Perfect separation of the dark and bright regions was obtained. Figure 2.8 was obtained using only the dark cluster as data. The dark cluster was split on the basis of position into the two clusters shown by points marked 1 and 2.

The gap between the dark and bright intensity values can be reduced by changing the value of sigma used in generating

## MODELING OF SHADOWS IN RADAR CLUTTER

the bright rayleigh data. To create a more demanding exercise for the algorithm, the data set was recreated with sigma equal to 50. Logarithmic scaling of the intensity values was again used. Figure 2.9 shows the result of clustering for light and dark intensity regions. Points in the dark cluster are indicated by ones, and points in the bright cluster are indicated by twos. Note that a number of bright points have been misclassified by being included in the dark cluster. It would be expected that statistical variation among the intensity values of the non-shadow data would lead to some "bright" points with intensity values that are in fact as low as the values of the intensities of the shadow data. The number of misclassified points shown in the figure corresponds roughly with the number that would be expected on statistical grounds.

Problems with convergence of the Forgy-Jancey algorithm were encountered for the first time in generating the clusters shown in Figure 2.9. To generate this figure the algorithm was halted using MINREL = 100; i.e. on the first pass for which fewer than 100 data entities were reassigned. Figure 2.10 shows the result of running the algorithm all the way to convergence, a process which required nearly five minutes of CPU time. It is interesting, and somewhat surprising, to note that the dark cluster of Figure 2.10 includes even more misclassified "bright" points than does the dark cluster of

## MODELING OF SHADOWS IN RADAR CLUTTER

Figure 2.9.

The dark cluster of Figure 2.10 was used as the data base for the second clustering pass, in which individual shadows were separated. Figure 2.11 shows the result of clustering by position with NC equal to two. In this case the two shadows were correctly separated, but each cluster consists of a dense core, corresponding to the actual shadow, surrounded by a sparse "halo" of points. In an attempt to separate the shadows from their halos, several clustering passes were made on the data with NC greater than two. Figure 2.12 shows the result of clustering by position when NC equals three. The points marked zero are excluded from the data set. The three clusters are indicated by points marked 1, 2, and 3, respectively. (Cluster two is in the upper right-hand corner of the figure.) Note that the actual shadows still form the cores of separate clusters, but that both are still surrounded by halos. Figures 2.13, 2.14, and 2.15 are the results of clustering with NC respectively equal to four, five, and six. Figure 2.16 shows the case with NC equal to seven. One of the shadows has been split into two clusters, marked 1 and 7 in the figure. This is an undesirable result, from which two lessons can be learned: first that the Forgy-Jancey algorithm will find as many clusters as it is instructed to, whether or not these clusters are meaningful, and second, that increasing the number of

## MODELING OF SHADOWS IN RADAR CLUTTER

clusters is not an effective method for removing the halo from a dense core of points.

Identifying the value of NC that corresponds to the number of clusters actually present is difficult when dealing with unknown data, especially data of high dimensionality that cannot be readily plotted. Jancey [1] suggests beginning with  $NC = 1$  and running the algorithm repeatedly with  $NC = 2, 3$ , etc. As the number of clusters increases, the value of the total summed deviations obtained at convergence will decrease. As "natural" clusterings are obtained, there should be significant drops in the value of the total summed deviations. An example of this effect is presented in [1]. Figure 2.17 shows the values obtained for total summed deviations in the above rayleigh example when  $NC = 1$  through 7. Note that the most significant drop occurs when  $NC$  is raised from one to two.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 2.3 Removal of Outliers

Fordon [9] suggests a modification of the Forgy-Jancey algorithm to allow rejection of data entities that are too far from any of the cluster centroids. A data entity can be isolated if it satisfies the relation

$$|D_{min} - D_{avg}| < (\text{OUTLYR}) \times D_{avg},$$

where  $D_{min}$  is the distance between the entity and the nearest cluster centroid and  $D_{avg}$  is the average of the distances between the data entity and each of the cluster centroids. The parameter OUTLYR is assigned a value between zero and one by the programmer. A value of zero results in no entities being isolated, while a value of one results in every entity being isolated. The main loop of the Forgy-Jancey algorithm was modified so that whenever an isolated entity is encountered it is designated as the centroid of a new cluster. This allows the program to increase the number of clusters. A provision was included to limit the number of clusters to a programmer-specified maximum, since CPU time was found to increase dramatically with the number of clusters. In interpreting the results of a clustering run, the programmer has the option of disregarding all newly-formed clusters as representing outliers; i.e. sets of anomalous entities that lie outside the known clusters, or of accepting all of the clusters in the expectation that the program has found a more natural

## MODELING OF SHADOWS IN RADAR CLUTTER

number of clusters than the programmer originally specified.

An example of a clustering run using the modified Forgy-Jancey algorithm is shown in Figure 2.18. The data set is the 100x100-point array depicted in Figure 2.3, with each entity having an intensity value of zero or one. Three outliers were created, having intensity values of zero, and lying outside the simulated shadows. The Forgy-Jancey algorithm was run twice, once to separate out the low-intensity points and once to cluster these points into individual shadows. Figure 2.18 is the result of the second run. To generate this figure, the program was started with NC = 2 and OUTLYR = 0.6. The program responded by creating two additional clusters, identified in the figure by the numbers 3 and 4. Although the program successfully located the three outlying points, it should be noted that the value used for the parameter OUTLYR is fairly critical. The program was unable to locate all of the outliers when run with OUTLYR = 0.4.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 2.4 Testing of Actual Radar Data

A tape containing a segment of a PPI display was provided by the Signal Processing Laboratory at RADC, courtesy of William L. Simkins, Jr. The tape contains data in essentially the same format as the simulated data described above: each data entity is comprised of a set of three components representing x and y position and intensity. The data were created by quantizing a photograph of the original CRT display, and in consequence the intensity scale of the data is compressed by the responses of the CRT phosphor and the film. Figure 2.19 depicts one quadrant of the radar data. The grey scale used in the figure consists of the symbols (W, \*, A, I, +, -, ., blank), in order from highest to lowest intensity. Figure 2.20 shows the low-intensity regions of the quadrant of data. The set of low-intensity entities was used as a data base for the Forgy-Jancey clustering algorithm. Numerous runs were made, using various values of input parameters. A typical output is shown in Figure 2.21. To generate this output twenty-one initial cluster centroids were chosen, corresponding roughly to the centers of the clusters visible in Figure 2.20. The outlier parameter was set to OUTLYR = 0.9, and the algorithm was stopped when fewer than 33 entities changed cluster membership (MINREL = 33). The program ran for nearly four minutes and generated a total of

## MODELING OF SHADOWS IN RADAR CLUTTER

twenty-five clusters.

Examination of Figure 2.21 reveals the twenty-five clusters, designated by the numbers zero through nine and the letters A through O. A close examination of this Figure reveals that many of the clusters found by the program do not correspond to the natural clusters evident in Figure 2.20. In particular, attention is called to cluster K, which is composed of the adjacent parts of four natural clusters, clusters C and O, which split a single natural cluster, cluster 2, which contains some entities that should properly belong to cluster 3, and cluster D, which encroaches on the territory of clusters C and O.

In processing the actual radar data, the Forgy-Jancey algorithm had a great deal of difficulty in locating the natural clusters. It is likely that this difficulty stems from two sources: the highly irregular shapes of the natural clusters, and the fact that the natural clusters vary greatly according to size. It is believed that the total-summed-deviations figure of merit used by the algorithm tends to emphasize the compactness of clusters. Thus whenever a cluster becomes too long and narrow, the algorithm will split it into two smaller clusters. The procedure used by the program for generating new clusters tends to result in clusters of about equal size. Hence the difficulty in dealing with a data set that contains clusters as small as cluster 7 and as

## MODELING OF SHADOWS IN RADAR CLUTTER

large as cluster 6.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 2.5 Conclusions

The Forgy-Jancey clustering algorithm is a powerful tool whose principle advantages seem to be its generality and simplicity. The generality results from the ability of the algorithm to handle data sets of essentially any size, consisting of data entities of any dimension. The number of clusters that the algorithm can handle is also essentially unlimited, and the programmer is free to prescribe arbitrary coordinate scaling and an arbitrary distance measure. The simplicity of the algorithm is helpful when modifications are needed. In carrying out the present study it was necessary to add several input and output routines, and to modify the main loop of the algorithm to incorporate the ability to generate new clusters.

The drawbacks to using the Forgy-Jancey algorithm are clearly illustrated in Section 2.4, particularly in Figure 2.21. The Forgy-Jancey algorithm is known to be capable of finding clusters that bear no relation to any natural clustering [10], and must be used with extreme care on unknown data that can not be visually inspected. Some of the difficulty in locating the natural clusters in Figure 2.21 can be attributed to the procedure used for generating new clusters. Without this procedure, however, it is necessary to specify the number of clusters in advance. Jancey's procedure

## MODELING OF SHADOWS IN RADAR CLUTTER

for monitoring the total summed deviations offers one method by which the number of natural clusters might be determined.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 3.0 MINIMAL SPANNING TREE ALGORITHM

#### 3.1 Introduction

In this section we describe the minimal spanning tree algorithm [2] for clustering data. This graph-theoretic method differs greatly from the Forgy-Jancey method described in the previous section. We used an implementation of the MST (minimal spanning tree) algorithm in our investigation obtained in the ARTHUR subroutine package. ARTHUR is a general package of data manipulation and analysis subroutines that was obtained from the University of Georgia. The MST subroutine in the package is called TREE and requires use of a distance subroutine DIST, plus input and output utility subroutines also contained in the package. ARTHUR will be discussed further in this section, along with the description and discussion of TREE. Before we examine how TREE clusters our data, we must first discuss the MST algorithm and how it can be used for clustering.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 3.2 Minimal Spanning Trees

A minimal spanning tree is an undirected graph that includes all the given data points, has no loops or cycles, and has a minimal distance over all the edges in the tree. The MST can use any distance metric to describe how close (or similar) the data points are, but the important point is that the resultant tree will have the smallest total distance of any of the possible spanning trees. The choice of the distance metric makes this method general for any type of data; however, in our study we only considered euclidean distance for x-y data. This is because we were only doing x-y clustering in this phase of our study. A different form of metric could be used for some other clustering, such as intensity clustering. The ARTHUR MST algorithm clusters data by finding "inconsistent" edges in the MST; the tree itself does not contain clusters, but the algorithm points out edges that when deleted would give a natural clustering. This method of forming clusters eliminates one of the major problems encountered in using the Forgy-Jancey algorithm, namely that we must tell it how many clusters to find, but the MST clustering algorithm then needs some form of information describing what an inconsistent edge is. The factors we use to determine whether an edge is inconsistent depend on the average length of the neighboring edges, where we can specify how deep along the tree a neighboring edge is. We

## MODELING OF SHADOWS IN RADAR CLUTTER

also need to know how many standard deviations from the average the edge is question is. The edge is cut based on these parameters, if the edge is more than the specified number of standard deviations from the average.

This method of forming clusters is general in as much as the distance metric is general and the clustering criterion is relative to distances in the tree. This method of choosing clusters based on inconsistant edges in the minimal spanning tree will work well on many types of data, as shown by Zahn [2]. After the tree is formed, the inconsistant edges are found and deleted and the connected groups that are left are the clusters. This means that we can form any number of clusters, not just a predetermined number or range of clusters. Also, we can find the tree just once, and try several different sets of clustering parameters on it, without the expense of finding the MST each time.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 3.3 ARTHUR and the MST

We used the TREE routine in ARTHUR to cluster data using a minimal spanning tree. TREE can be given three parameters, the DEPTH of the neighbors, a FACTOR used in finding inconsistancies, and a SPREAD of the number of standard deviations allowed before an edge is inconsistent. An edge is inconsistent if its length is greater than FACTOR times the length of the neighboring edges, or if it is more than SPREAD standard deviations longer than the average of the neighboring edges, where neighboring edges are all edges within DEPTH edges from the edge being tested [7]. A copy of the ARTHUR documentation is contained as an appendix in Fordon [9]. In using TREE, the data must be input to ARTHUR using one of its utility routines. The distance matrix containing the distance between every pair of data points must be found and stored using the DIST utility routine, and then TREE is called. Part of a test run of ARTHUR and TREE is shown in Figure 3.1, showing how TREE outputs the MST as a list of nodes and neighbors. The resulting clusterings are shown in Figures 3.2, 3.3, and 3.4, as lists of entities (they call them patterns) belonging to the various clusters. These clusterings were all done on the same MST in the same run of ARTHUR; recalculation of clusters with different parameters does not require recalculation of the distance matrix or the MST. The main

## MODELING OF SHADOWS IN RADAR CLUTTER

limitation is that in examining  $N$  data entities, we require on the order of  $N^2$  units of storage for the distance matrix, and of course the amount of time to compute all the distances. This is significant in that if we double our number of points, we need four times the storage and computer time to do the same processing. For example, a 10 by 10 picture has 100 points, and needs 10,000 words of storage for its distance matrix; if we increase the picture size to 100 by 100, we have 10,000 points and need 100,000,000 words to store the distance matrix. So to enlarge a picture by a factor of 10 requires a  $10^4$  increase in memory.

A temporary limitation of TREE is that it only prints a description of the MST in terms of the nodes and edges as in Figure 3.1, and clustering cannot be displayed other than in a list of cluster members. The cluster membership is not output in an ARTHUR format file so we cannot display the clustering using one of ARTHUR'S plotting routines.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 3.4 Actual Clustering Using TREE

In order to test the clustering ability of TREE, we used data previously used in the Forgy-Jancey testing. We took part of one of our rayleigh data files, namely cluster number 2 from Figure 2.11, and converted it to ARTHUR data format. This picture consists of 218 data entities, shown as 1's in Figure 3.5. We want to isolate the obvious group of 121 entities from the surrounding outlier entities. We ran ARTHUR on this data, found the distance matrix using euclidean distance, and called TREE. TREE found the minimal spanning tree, shown in Figure 3.6, and clustered the data according to several different sets of parameters. The clustering formed by the default parameters of DEPTH=3, FACTOR=2 and SPREAD=0, as in Figure 3.2, did eliminate some of the outliers from the desired cluster, but left 207 entities in the cluster, which is not very close to our desired 121 entities in the cluster. These parameters did not form many extra clusters; only five clusters were found, separating only 11 outlier entities from the main cluster.

By trying only a few different combinations of parameters, we made the clustering much better. Using DEPTH=1, FACTOR=1, and SPREAD=1, we found 33 clusters, shown in Figure 3.3. This clustering put 132 entities in the main cluster, eliminating almost all the outliers. It removed 86 of a possible 97 outlier entities from the main cluster. Using the same

## MODELING OF SHADOWS IN RADAR CLUTTER

parameters as before, but setting SPREAD=0.5, we obtained the best results as shown in Figure 3.4. We found 37 clusters, and the main cluster had only two outlier points included in it, for a total of 123 points in that cluster. We were therefore able to eliminate 95 of 97 outlier entities. This clustering is drawn out in Figure 3.7, shown as the remaining edges in the tree, with all of the clusters circled to identify them.

This clustering differs from possible clusterings by the Forgy-Jancey algorithm in several ways. First, it cannot split up dense clusters because it attempts to cut edges only where they are not dense. Also, since this is a single pass clustering, the decisions are made once, not iteratively updated. This means that there are no convergence problems with the resulting waste of computer time. Again, since we need not choose in advance how many clusters to find, we eliminate the problem of re-running the program with different numbers of clusters until an optimal number is found. Instead, the program will find the appropriate number of clusters, given the parameters for cutting inconsistent edges.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 3.5 Conclusions

The TREE routine can create clusters from noisy data, as shown in Figure 3.7, without problems such as the splitting of desired clusters found with the Forgy-Jancey clustering. However, TREE in its present form is relatively expensive in both time and storage required. Part of this expense is due to the overhead involved in using the ARTHUR subroutine package, but using ARTHUR is worthwhile because it gives us a powerful and flexible system of programs, rather than a stand-alone MST program. ARTHUR provides us with its flexible input routine, and has built-in routines to do scaling on the data, and even can orthogonalize data that is not orthogonal. It also has a histogram plotter to check the data distribution, three different data plotting routines, and several classification subroutines besides TREE, such as a bayesian classifier and a hierarchical classifier. A stand-alone MST program might be more efficient, but would take substantial programming effort and would sacrifice all those valuable subroutines. The space

2

expense is due to the  $N^2$  size of the distance matrix, which presently limits maximum size of the data set that can be analyzed. However, ARTHUR can store data in both memory and disk dynamically by swapping parts of the data set between core and disk, saving memory at the cost of more disk I/O operations.

## MODELING OF SHADOWS IN RADAR CLUTTER

TREE is more flexible than Forgy-Jancey because it does not have to told how many clusters to find. In addition, Zahn [2] suggests a variety of alternative schemes for dividing a minimum spanning tree into clusters, each of which might be useful for a different pattern of data entities. If some of these schemes were to be incorporated into the MST subroutine, it would significantly enhance the usefulness of the minimum spanning tree as a clustering technique.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 4.0 PREPROCESSING TECHNIQUES

#### 4.1 Introduction

In this section we describe a few preprocessing techniques used on the data before applying them to the clustering algorithms discussed in the preceding sections. The preprocessing techniques used are:

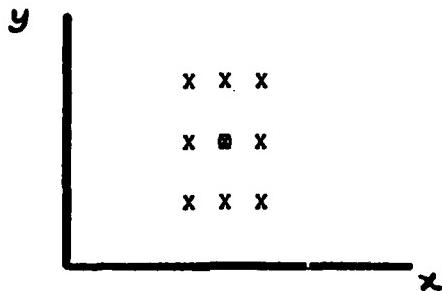
- a) point-to-point local variance
- b) point-to-point local gradient
- c) radial derivative

Of all these preprocessing techniques the radial derivative is the most promising one. These techniques were applied to the compressed digitized radar data provided to us by RADC and shown in Figure 2.19. In each case we applied the technique and examined the output data for clues for identification of bright and dark areas. Results obtained for each technique are discussed individually in the following. For each of these techniques we used half of the picture provided as otherwise the amount of data to be processed was too large.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 4.2 Point-to-Point Local Variance

The idea behind this method is that probably the variation of the intensity of the radar return will be very small inside a shadow and it will be large in the peripheries of shadows. For each point in the picture, variance was calculated by considering nine neighboring points on the left-right and above and below this point as shown below



We first calculated the average intensity  $\bar{I}$  for these nine points. Then we calculated the local variance by using the formula

$$\text{Variance} = \sum_{j=1}^9 (I_j - \bar{I})^2 / 9$$

where  $I_j$  indicates intensity at the  $j$ -th point.

Then these variances were plotted on an eight level grey scale using the following symbols: (W, \*, A, I, +, -, ., blank) in descending order of the value of variance (W represents the highest and blank represents the lowest values of the

## MODELING OF SHADOWS IN RADAR CLUTTER

calculated variances). This plot of the variances is shown in Figure 4.1.

An examination of this figure reveals that the shadowy areas do have small variances and the edges have large variances. However, some areas with constant but high-intensities also have small variances. Hence it appears that the local variance alone is not sufficient for clear cut identification of shadows. Other information like intensity and edge information may have to be considered together with the local variance. Further, with this local variance method all directional information i.e. the variation of intensity with distance is lost.

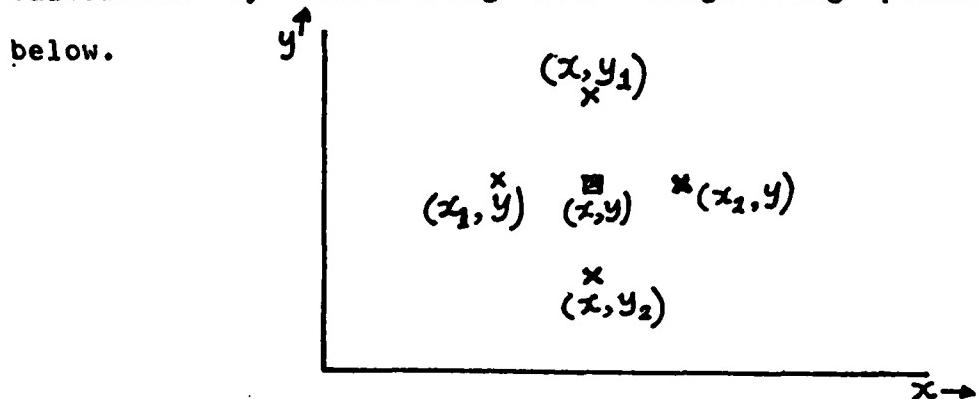
For an 128 x 128 point data set computation of local variances needed 33 seconds of CPU time. The storage

requirement was  $2 \times 128^2$  bytes,  $128^2$  bytes for the original array and  $128^2$  bytes for the calculated variances. We used 200 kilobytes of storage and it was sufficient. Expense for one run was \$2.22.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 4.3 Point-to-Point Local Gradient

The idea behind using local gradient was that there should be higher variation of intensity on the edges of shadows than inside them. For each point of the array local gradient was calculated by considering four neighboring points as shown below.



By using the formula,

$$\text{local gradient} = \sqrt{(I_{(x_2,y)} - I_{(x_1,y)})^2 + (I_{(x,y_2)} - I_{(x,y_1)})^2}$$

where  $I$  represents the intensity at the point  $(x, y)$ . The calculated local gradients were plotted over an eight level grey scale using the same symbols as in the case of local variance. This plot is shown in Figure 4.2.

An examination of Figure 4.2 shows that in the case of local gradient also shadows and high constant intensity areas have low gradients. The edges, however, showed larger gradient. Hence the local gradient above also can not be used to identify shadows conclusively. Other information like the

## MODELING OF SHADOWS IN RADAR CLUTTER

intensity, needs to be considered together with gradient.

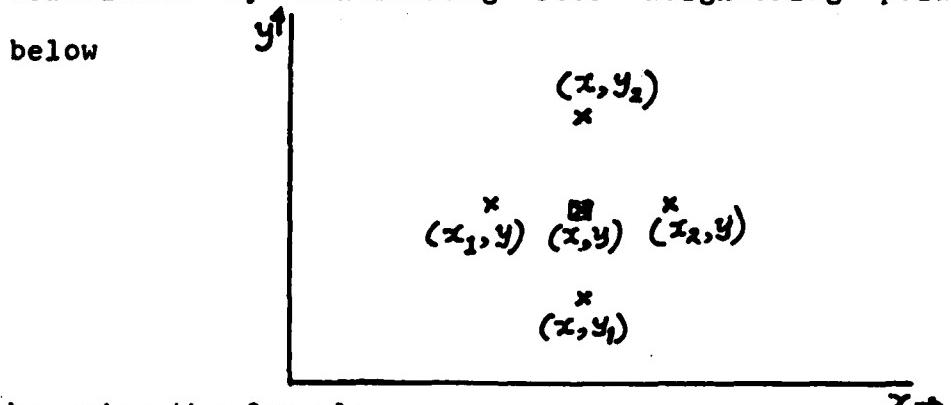
For a  $128 \times 128$  array of data points it took 25 seconds of CPU time to compute gradients. Storage requirement was also  $2^{12} \times 128$  bytes. An allocation of 200 kilobytes was sufficient. Expense for one run was \$1.91.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 4.4 Radial Derivative Method

In this method derivatives were calculated along radial lines along those of the radar beam starting from the location of the radar as shown by the X in Figures 4.3 and 4.4. The idea behind using radial derivatives is that as the radar beam travels in radial directions, the variation of intensity in the radial direction will be a better measure of the reflective nature of the objects illuminated by the radar. Also any object intercepting the radar beam will produce a shadow in the radial direction.

For each point of the data array radial derivative was calculated by considering four neighboring points as shown below



by using the formula

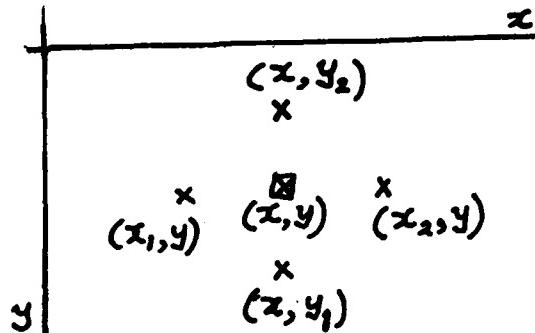
$$\text{Radial Derivative} = \frac{(I_{(x,y_2)} - I_{(x,y_1)})|y| + (I_{(x_2,y)} - I_{(x_1,y)})|x|}{\sqrt{2(x^2+y^2)}}$$

for the points in the first quadrant. This formula has to be

## MODELING OF SHADOWS IN RADAR CLUTTER

modified for other quadrants.

For example for the points in the fourth quadrant as shown, the radial derivative is given by



$$\text{Radial Derivative} = \frac{(I_{(x,y_1)} - I_{(x,y_2)})|y| + (I_{(x_2,y)} - I_{(x_1,y)})|x|}{\sqrt{2(x^2+y^2)}}$$

These radial derivatives were plotted on a seven level grey scale using the following symbols.

- # < -30
- 30 = < θ < -15
- 15 = < - < -5
- 5 = < Blank < 5
- 5 = < + < 15
- 15 = < A < 30
- 30 = < W

the plots are shown in Figures 4.5 and 4.6.

An examination of Fig. 4.5 reveals that:

1. All dark regions have low radial derivative

## MODELING OF SHADOWS IN RADAR CLUTTER

2. Range marker showed up.
3. Shadows were marked by the following characteristics: in front of the shadows we observed regions of high radial derivatives (cause of shadow) followed by regions of low radial derivatives (edges of shadow) further followed by regions of lower radial derivatives (shadow). It was also observed that shadows were followed by regions of higher (increasing radial derivative (elevated region beyond the shadow)). We also observed regions of negative radial derivative before shadows and regions of positive radial derivatives after shadows.
4. For uniform, but high intensity regions (non-shadows) we observed regions of positive radial derivatives before them and regions of negative radial derivative after them. Of course, these uniform intensity areas themselves have low radial derivative like shadows. Thus the signs of radial derivative before and after a region of low radial derivative may be useful in distinguishing shadows from other low derivative regions.
5. However, we found some exceptions to these observations also. Radial derivative seems to be the most promising of all the preprocessing techniques test by us for detection and identifying shadows.

However, in order to put confidence into the use of radial derivative for detection and identification of shadows, more testing with data with a corresponding topographical map is

## MODELING OF SHADOWS IN RADAR CLUTTER

necessary. We did not have the topographical map to verify our conclusions. Also, the radial derivative makes use of radial changes and it does not take into account of side-to-side changes. Using both radial and local derivatives may improve the decision making.

Originally we plotted the radial derivative information on a seven point grey scale. However this plot looked too cluttered with symbols. Hence to have a clearer picture we replotted the radial derivatives on a five level grey scale. The derivative values represented by +, blank, and - were replaced by blanks. This plot is shown for the upper quadrant in Figure 4.7. The regions of low intensity depicted in Figure 2.20 are shown dotted in Figure 4.7. Using the plot we examined the regions of high but relatively constant intensity preceding shadows to test them as precursors of shadows. We noticed that for these regions of high but constant intensity before shadows there were regions of positive derivative before them and regions of negative derivatives after them. This indicated that these regions are some sort of precursors of shadows. (These areas were selected by hand. This selection may be done by a computer program. However these programs may be quite expensive and it is easier to select them by the sight.) Figure 4.8 shows the same information as Figure 4.7, but with the precursors indicated by numbers 0-9 and letters

## MODELING OF SHADOWS IN RADAR CLUTTER

B-U.

Then we tried to distinguish these groups of bright regions to see if these belong to more than one type of regions. Our investigation did not bring out any significant differences among the various groups.

Next we attempted to correlate these precursor regions to the shadows. We did observe that there were shadows following many of these precursors. This gave confidence to our conclusions.

Finally, we used the spanning tree algorithm to cluster the dark areas (possible shadows). We did not obtain any accurate division of these areas into individual shadows.

For a 128 x 128 point picture it took 26 seconds of CPU time to calculate the radial derivatives. Storage requirement was  $2 \times 128^2$  bytes. We were allotted 200 kilobytes and it was sufficient. Expense was \$1.95 per run.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 4.5 Conclusions

Out of the three techniques of preprocessing used, the radial derivative technique seems to be the most promising for detection and identification of shadows. However, combinations of these techniques may prove to be more useful under some conditions. It is necessary to have a topographical map corresponding to the data used to verify our conclusions.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 4.6 Note

We, at Clarkson, wrote, debugged and tested the programs used for the grey-scale plots, calculation of local variances, local gradient and radial derivatives. These programs are specifically written for these types of data (type of numbers, data x-y, rather than radial format). These programs are now available on disk and also on tapes for further use on this type of project.

## MODELING OF SHADOWS IN RADAR CLUTTER

### 5.0 CONCLUSIONS

This report has presented the results of a comparison of nonparametric clustering techniques and their use in examining radar clutter from shadowed terrain. Successful techniques for extracting features from clutter will aid in understanding the underlying structure of the clutter and in turn will lead to the development of useful clutter models. The Forgy-Jancey and minimum spanning tree algorithms were tested on both simulated and actual radar data. Some preprocessing techniques were also tested on the actual radar data in connection with the minimum spanning tree algorithm. The Forgy-Jancey algorithm was found to be a very flexible tool, handling large multidimensional data sets with minimal cost in CPU time and storage. Unfortunately the performance of this algorithm was found to be inconsistent in terms of finding the "natural" number of clusters. In general, repeated applications of this algorithm are necessary in order to find the optimum number of clusters and parameter values. A more detailed discussion of the conclusions pertaining to the Forgy-Jancey algorithm can be found in Section 2.5.

The minimum spanning tree algorithm was found to be more robust in its ability to detect a natural clustering; it was

## MODELING OF SHADOWS IN RADAR CLUTTER

never observed to split a natural cluster as was the Forgy-Jancey algorithm. The ARTHUR implementation of the minimum spanning tree provides flexibility in clustering, since once the tree is found, the data can be clustered repeatedly without recomputing the tree. Finding the tree, however, turned out to be costly for large data sets, since the ARTHUR implementation requires calculation and storage of the distance between every pair of data entities. Alternative procedures for computing the tree are available, however, as are alternative procedures for clustering the data from the tree. A detailed discussion is presented in Section 3.5.

A discussion of the preprocessing techniques examined is presented in Section 4, with the conclusions in Section 4.5. The preprocessing was motivated by the need to reduce the amount of data provided to the minimum spanning tree algorithm, but was found to be a useful technique in its own right for extracting features from clutter. Computation of the radial derivative was the most revealing of the techniques studied for locating and defining the edges of shadows.

Clustering techniques appear to hold great promise for examination of the structure of clutter returns, provided that the techniques are applied with some care and common sense. A primary result of the work to date has been installation of the relevant programs on the Clarkson College computer, and the

## MODELING OF SHADOWS IN RADAR CLUTTER

familiarization of the researchers with their use. The next step would be a detailed application of these programs to actual radar data, including an examination of a wider variety of clutter properties.

## MODELING OF SHADOWS IN RADAR CLUTTER

### APPENDIX: PROGRAM LISTINGS

This appendix contains the listings of the programs developed and used in this project. Figure A.1 is the comments section from subroutine EXEC, and describes the input format for Forgy-Jancey data. Figure A.2 is the input specification for the outlier Forgy-Jancey subroutine EXEC3. Figure A.3 is the subroutine EXEC to go along with the comments in Figure A.1. Figure A.4 is subroutine EXEC2 which allows outputting of the sorted data. Figure A.5 is subroutine EXEC3, the outlier calling subroutine. Figure A.6 (Parts 1-3) is subroutine KMEAN, the original Forgy-Jancey clustering routine. Figure A.7 (Parts 1-3) is outlier subroutine KMEAN2. Figure A.8 is the RESULT subroutine, which outputs the results of KMEAN. Figure A.9 is subroutine RSULT2, which allows output of the sorted data array. Figure A.10 shows the driver routine MAIN, function DIST, and subroutines USER and USROUT for a sample Forgy-Jancey clustering run. Figure A.11 shows MAIN and DIST for a sample outlier run, and also another version of DIST used in intensity clustering. Figure A.12 is subroutine RAY, our rayleigh random number generator. Figure A.13 is the program that created the rayleigh data set with sigma = 10,000. Figure A.14 is the program that created the rayleigh data set with sigma = 50. Figure A.15 is the plotting program that produced

## MODELING OF SHADOWS IN RADAR CLUTTER

all of our 100 x 100 and 128 x 128 point cluster plots. Figure A.16 is the grey scale plotting program used to generate all grey scale plots without overprinting. These programs were all either written at Clarkson, or installed and modified at Clarkson to suit the purposes of the project. For a description of ARTHUR, see Section 3, and the ARTHUR program documentation [7].

## MODELING OF SHADOWS IN RADAR CLUTTER

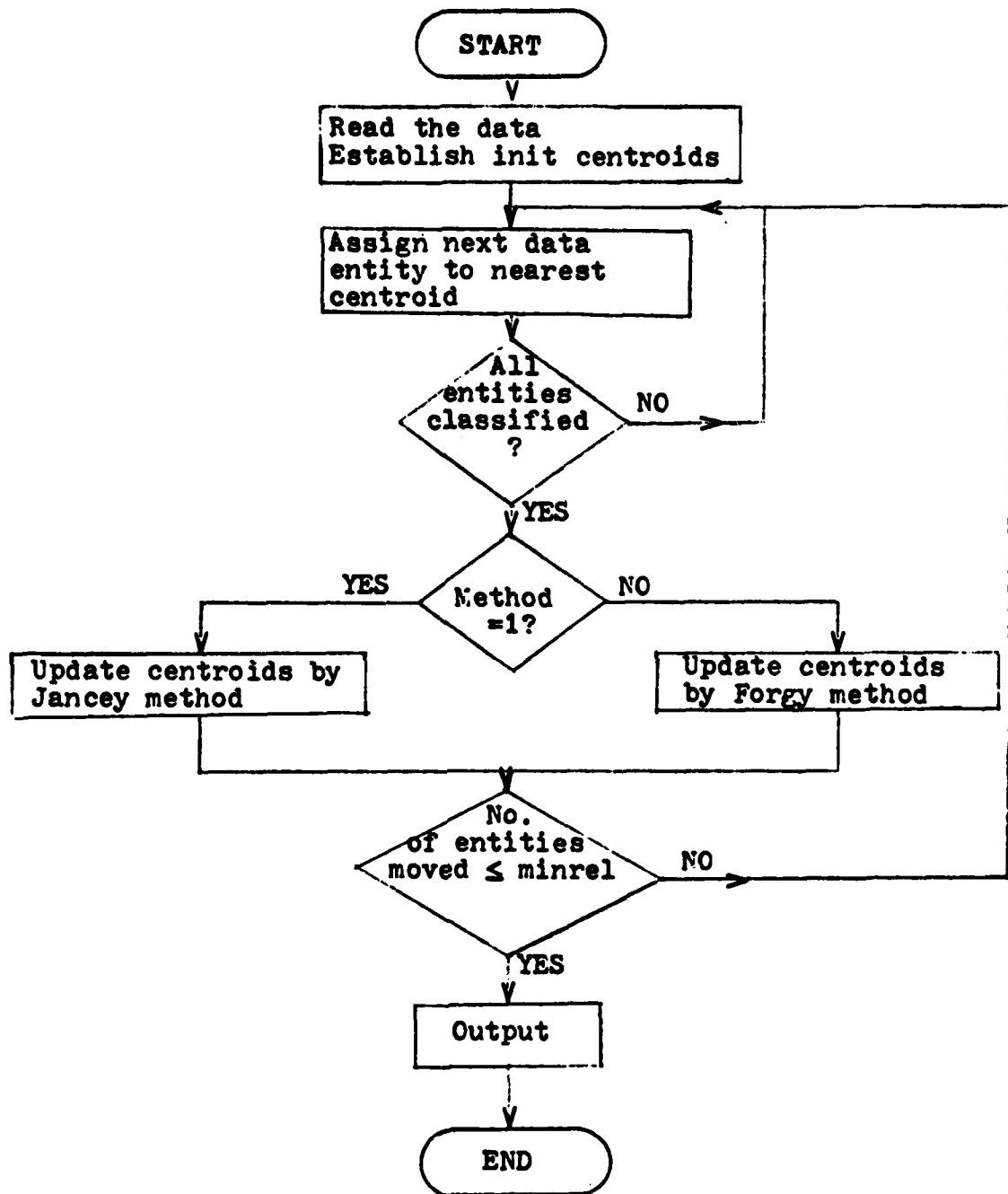


Figure 2.1: The Forgy-Jancey Algorithm

## MODELING OF SHADOWS IN RADAR CLUTTER

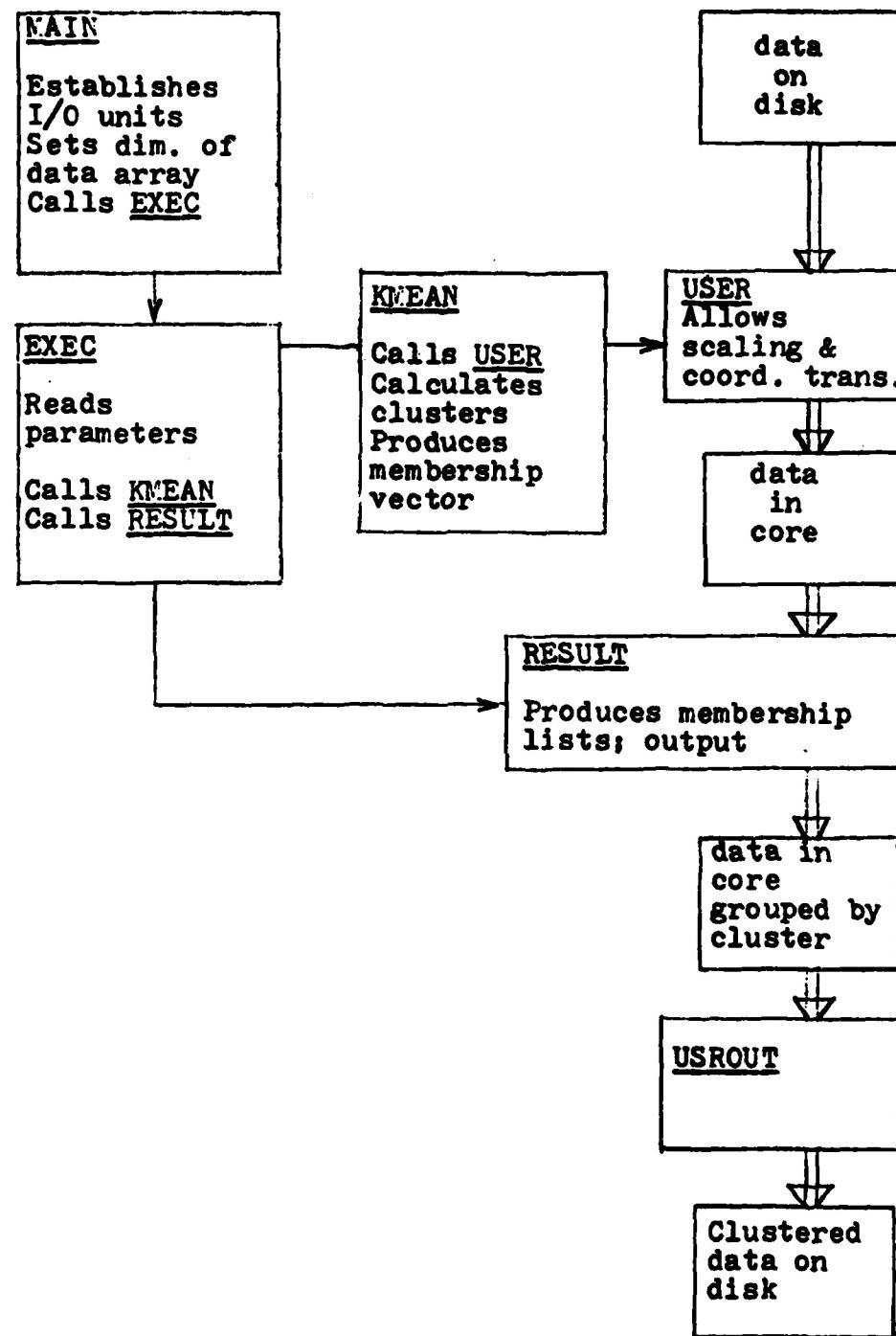


Figure 2.2: Forgy-Jancey Program Path

## MODELING OF SHADOWS IN RADAR CLUTTER

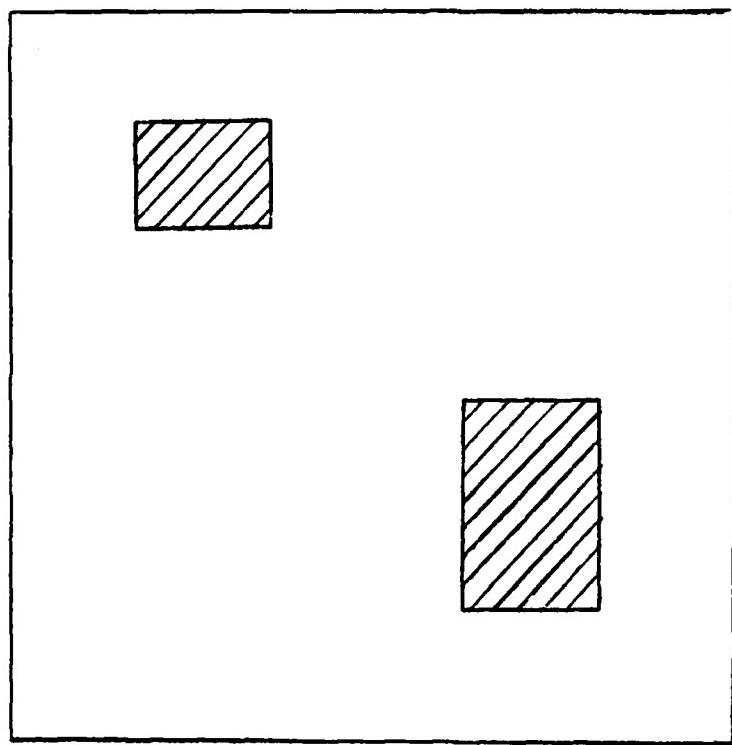


Figure 2.3: Test Scene for Clustering

MODELING OF SHADOWS IN RADAR CLUTTER

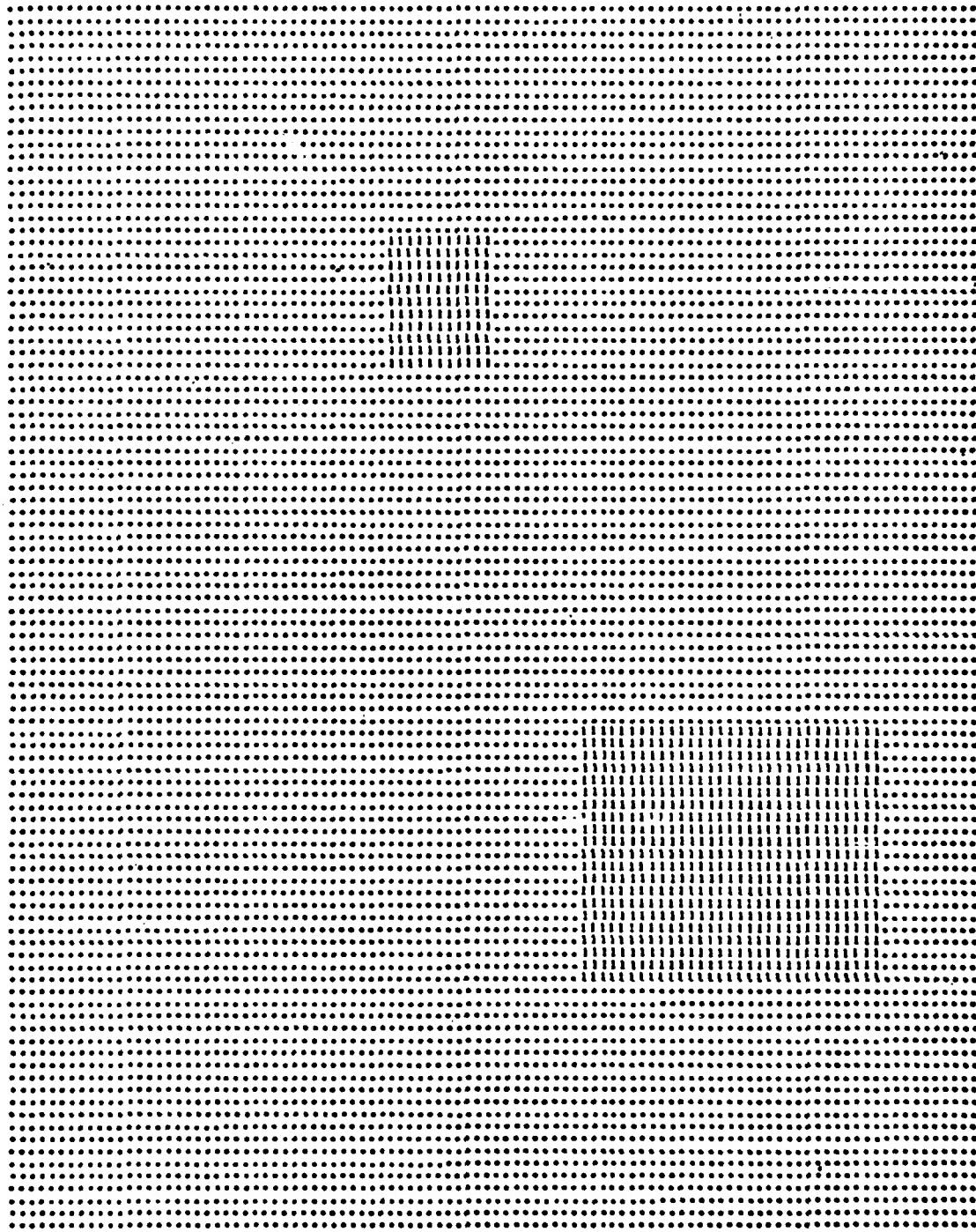


Figure 2.4: Intensity Clustering of Test Scene

## MODELING OF SHADOWS IN RADAR CLUTTER

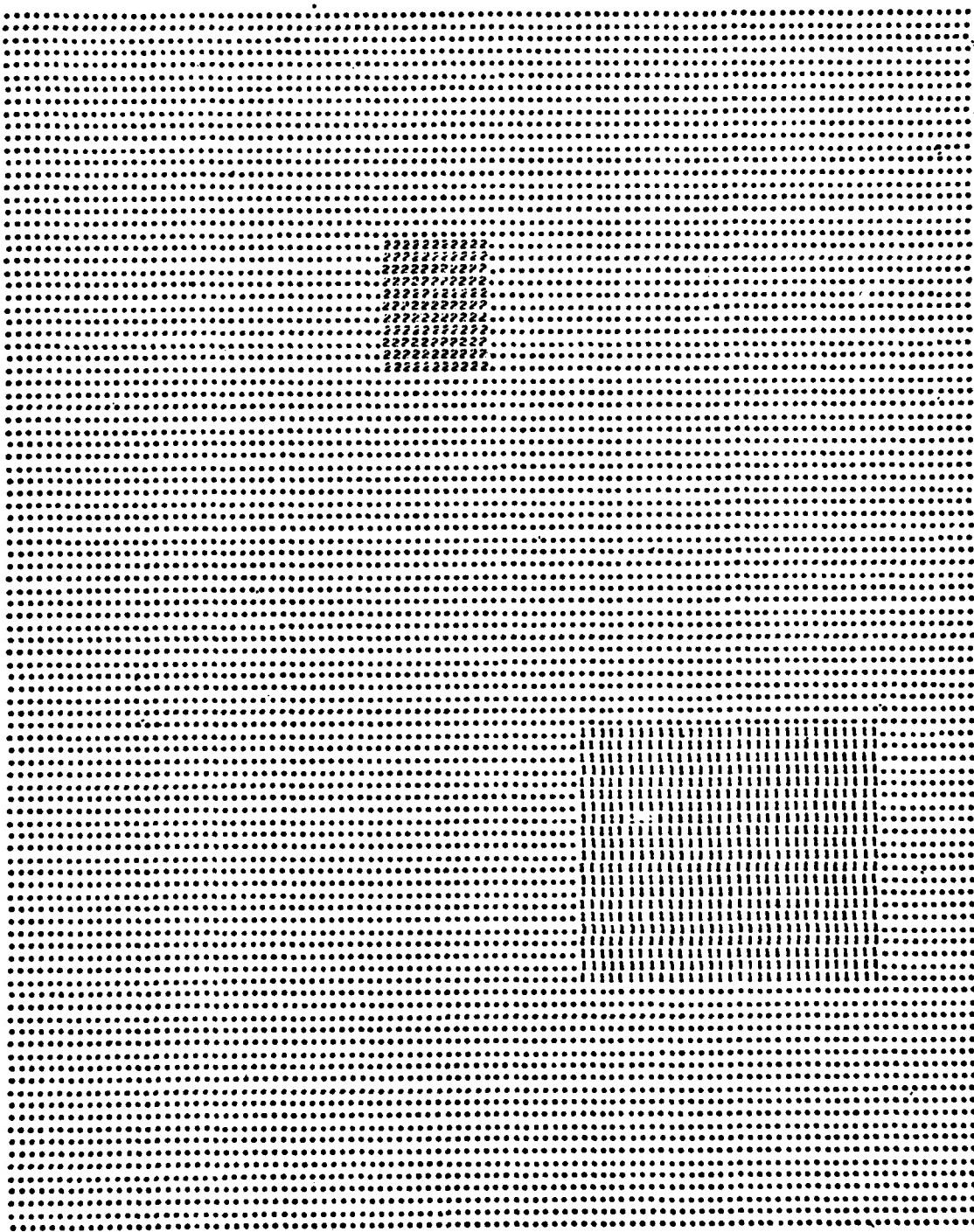


Figure 2.5: x-y Clustering of Test Scene

## MODELING OF SHADOWS IN RADAR CLUTTER

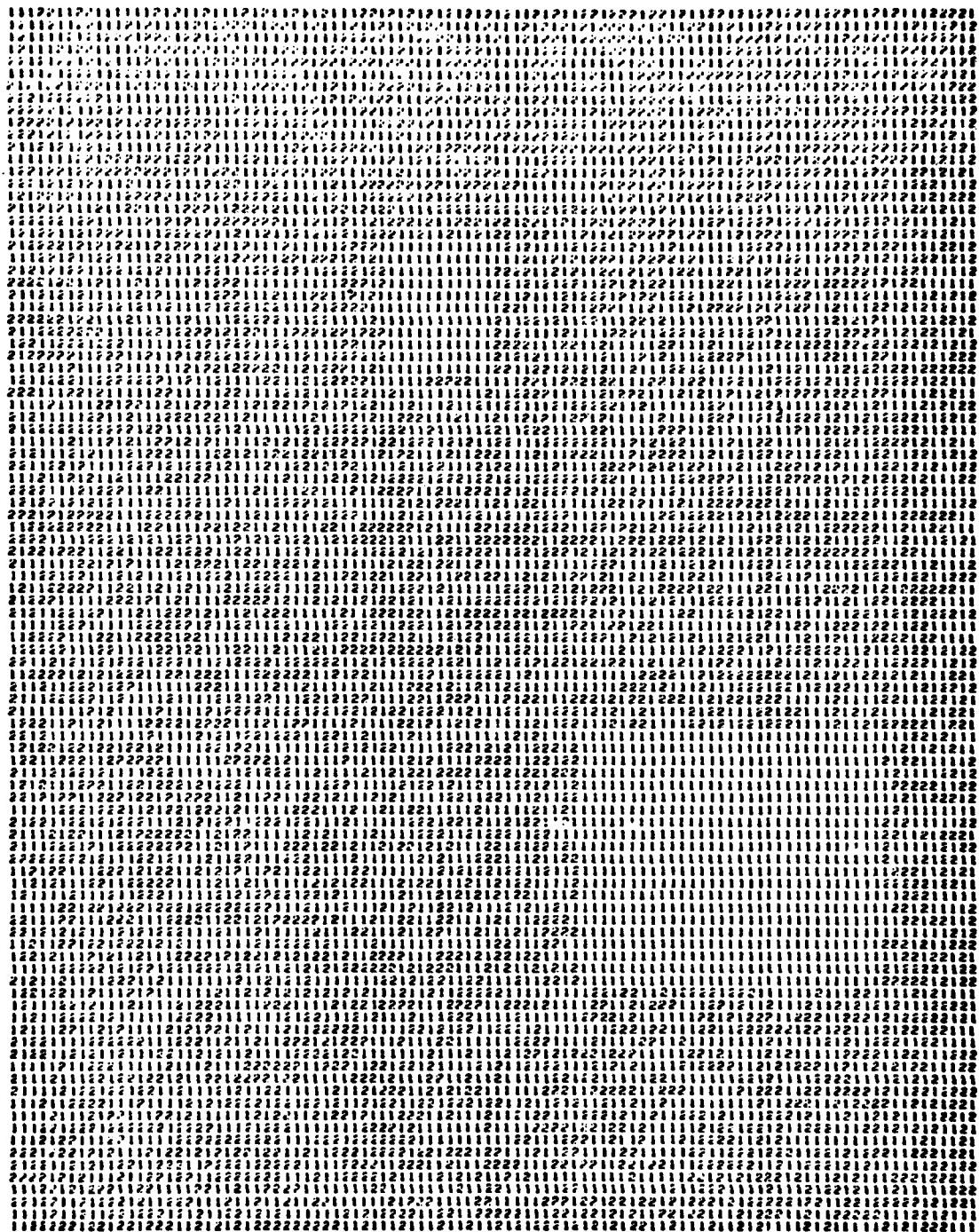


Figure 2.6: Intensity Clustering of Simulated Shadows

## MODELING OF SHADOWS IN RADAR CLUTTER

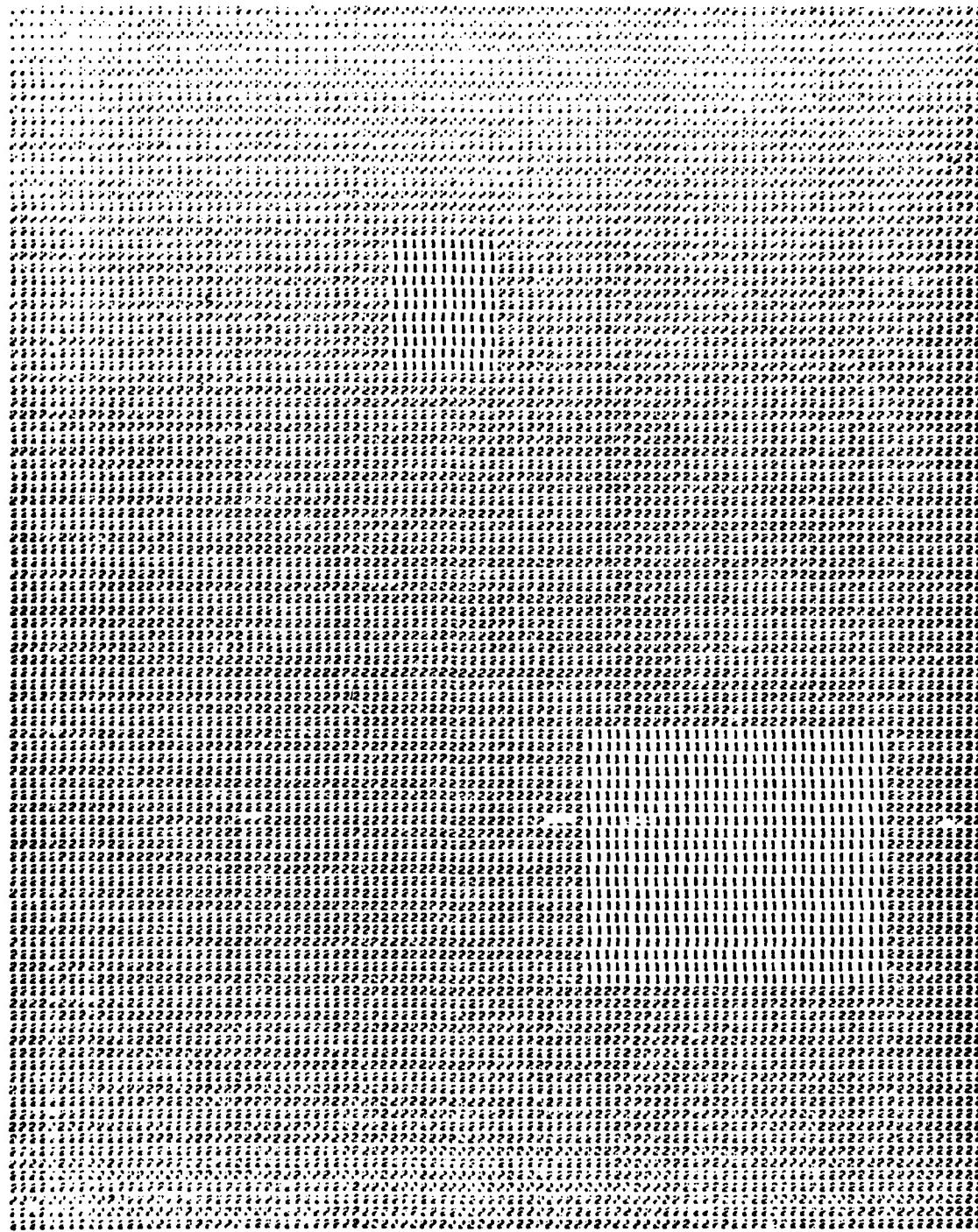


Figure 2.7: Intensity Clustering; Intensity in dB

## MODELING OF SHADOWS IN RADAR CLUTTER

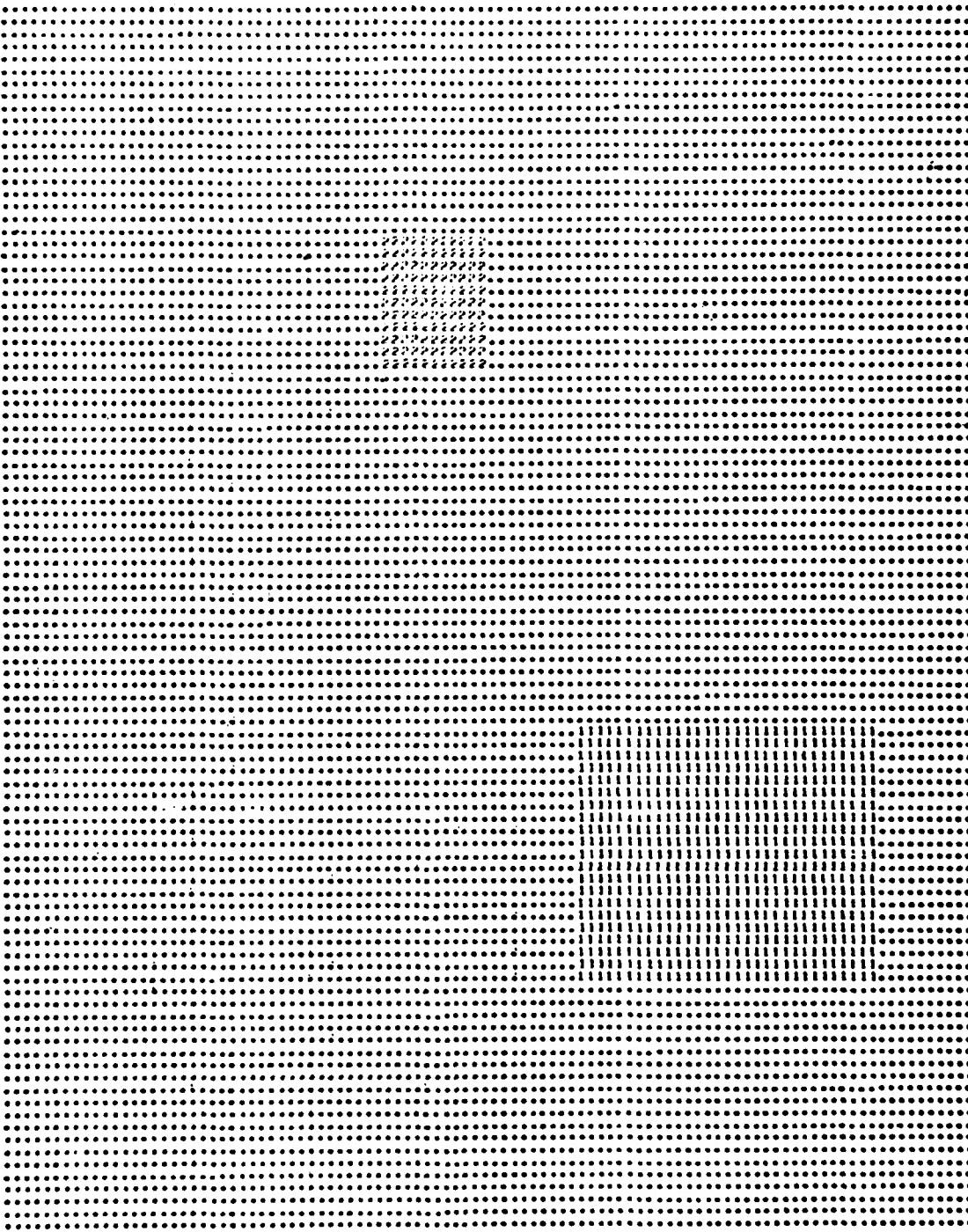


Figure 2.8: x-y Clustering of Simulated Shadows

## MODELING OF SHADOWS IN RADAR CLUTTER

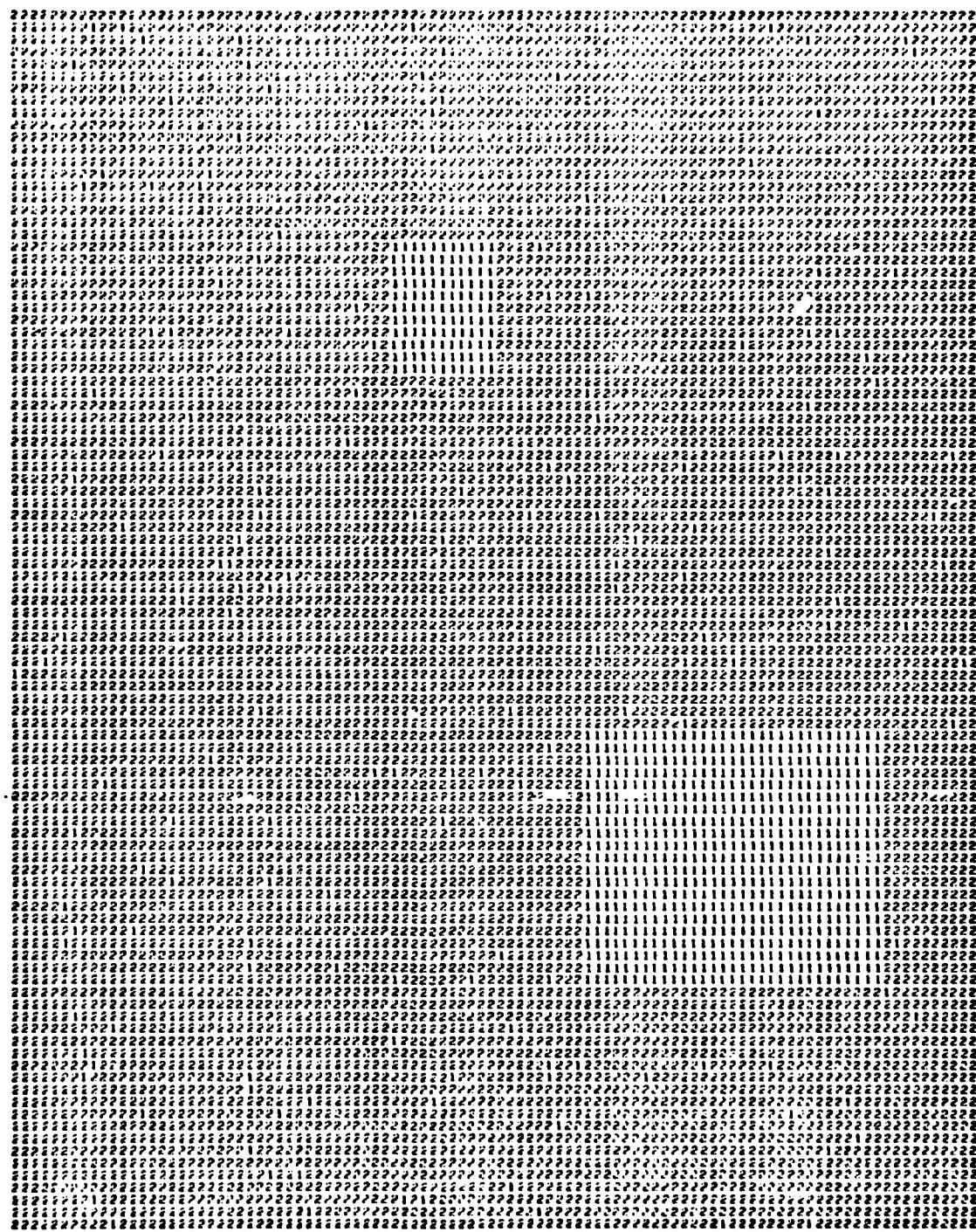


Figure 2.9: Intensity Clustering; Low Background Level

## MODELING OF SHADOWS IN RADAR CLUTTER

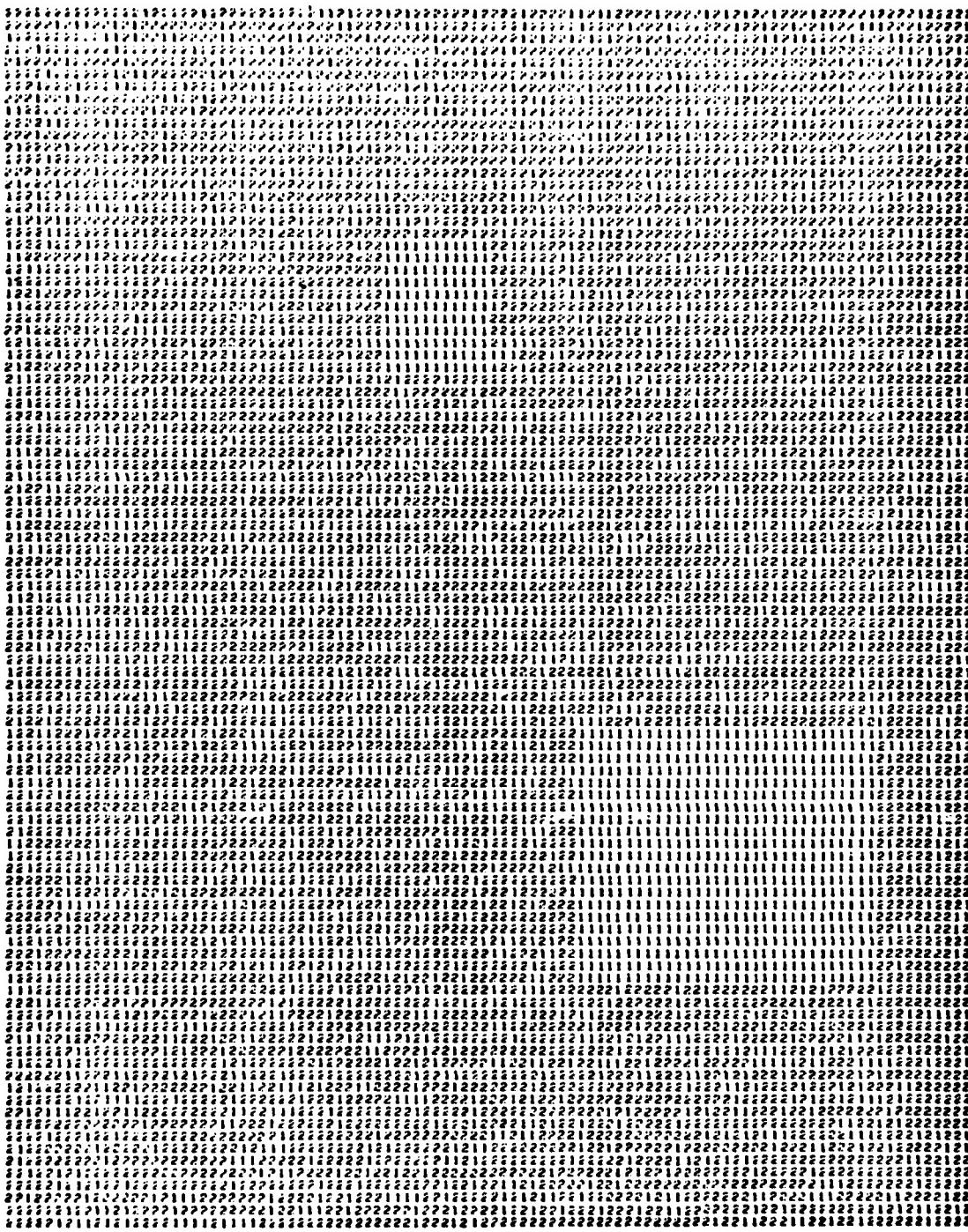


Figure 2.10: Low Background Level Run to Convergence

## MODELING OF SHADOWS IN RADAR CLUTTER

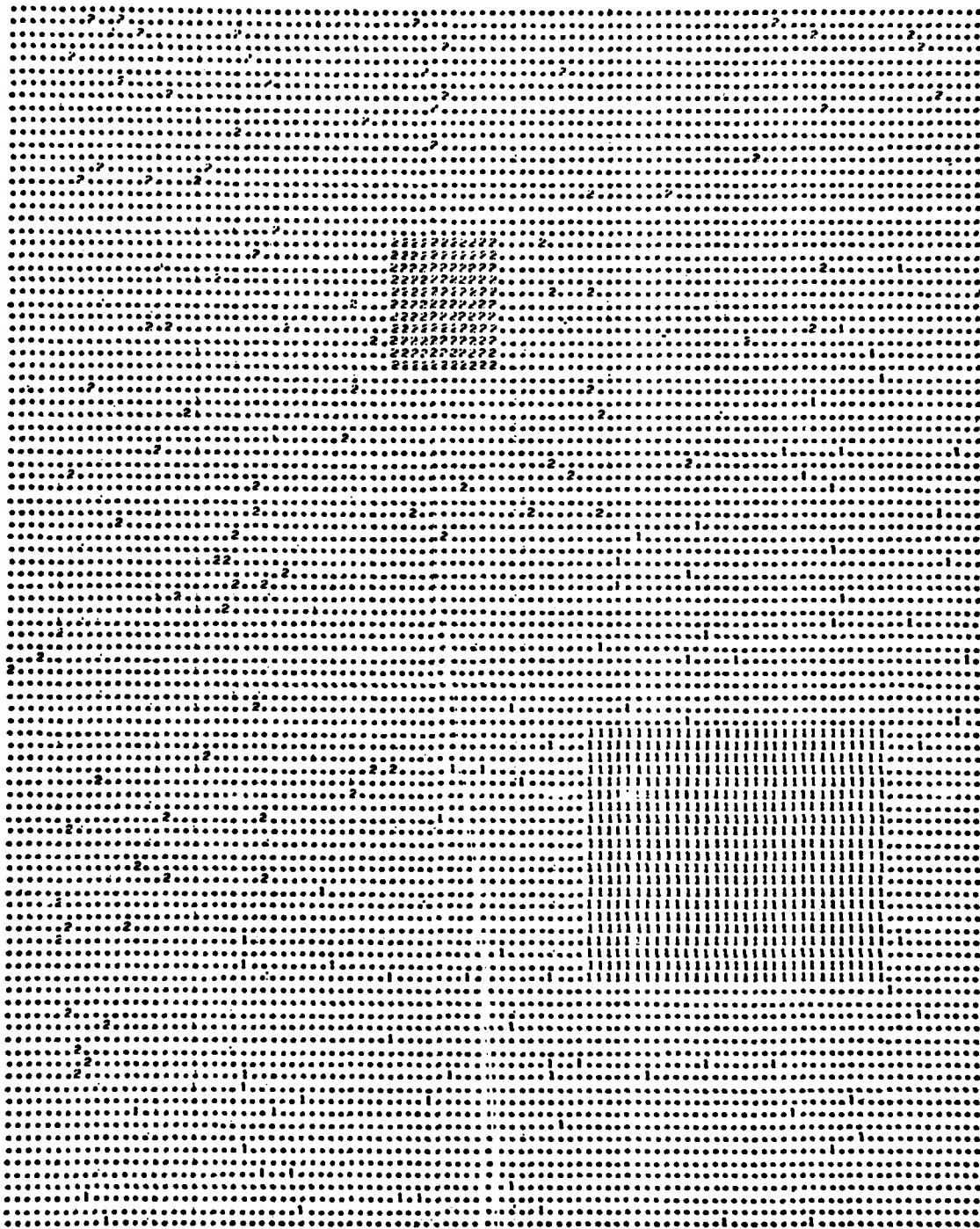


Figure 2.11: x-y Clustering, Low Background Level; NC=2

## 2.5 MODELING OF SHADOWS IN RADAR CLUTTER

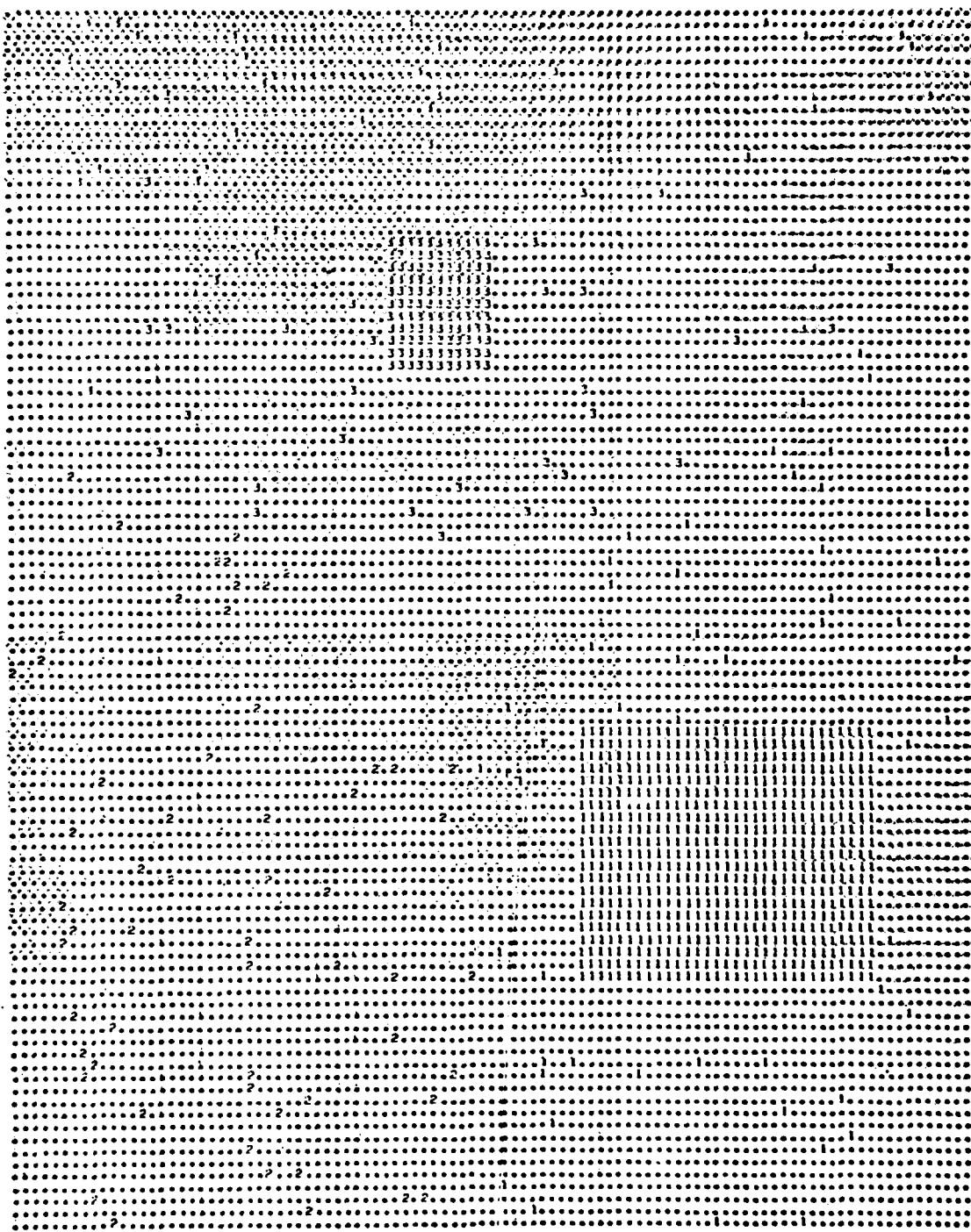


Figure 2.12: x-y Clustering, Low Background Level; NC=3

## MODELING OF SHADOWS IN RADAR CLUTTER

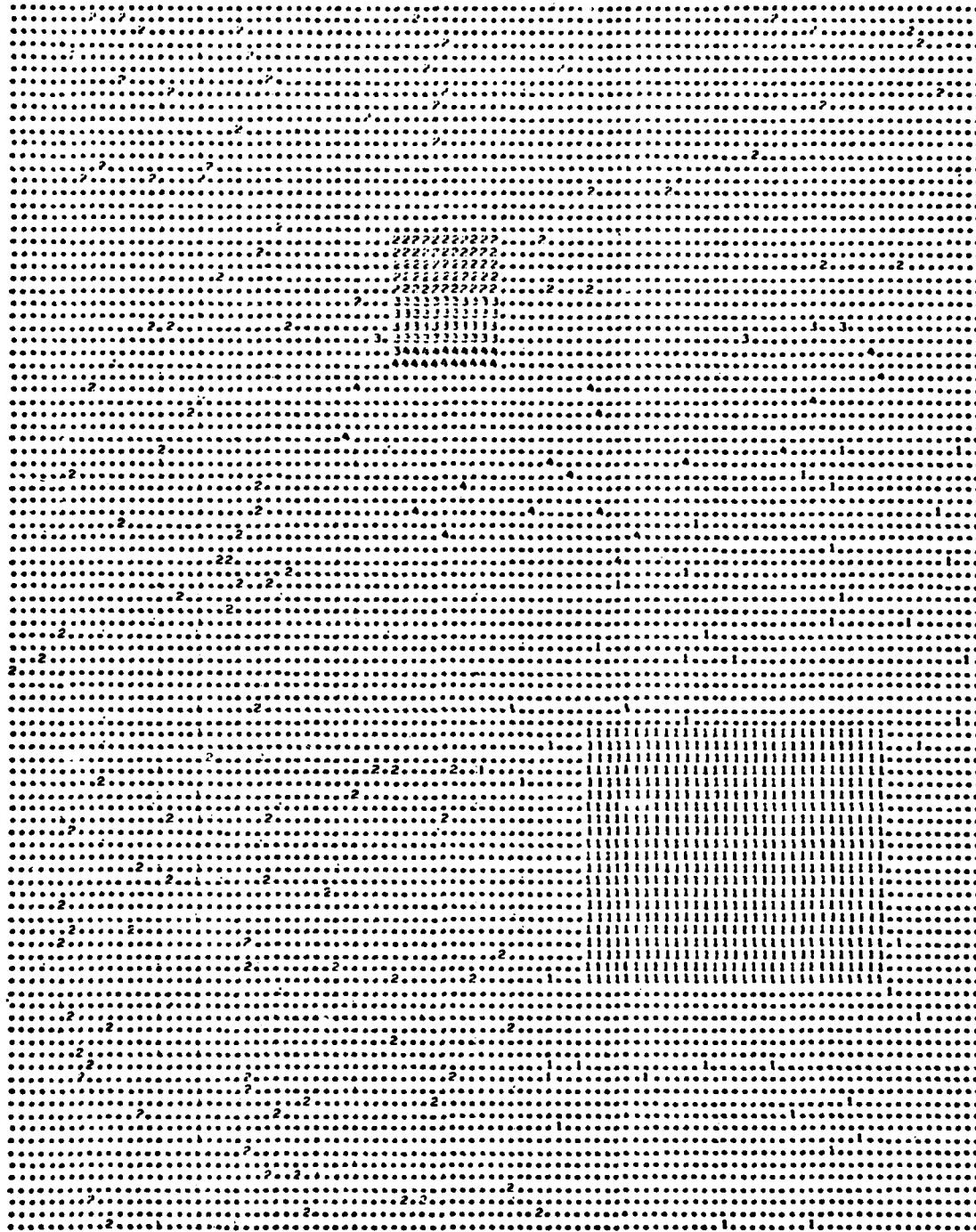


Figure 2.13: x-y Clustering, Low Background Level; NC=4

## MODELING OF SHADOWS IN RADAR CLUTTER

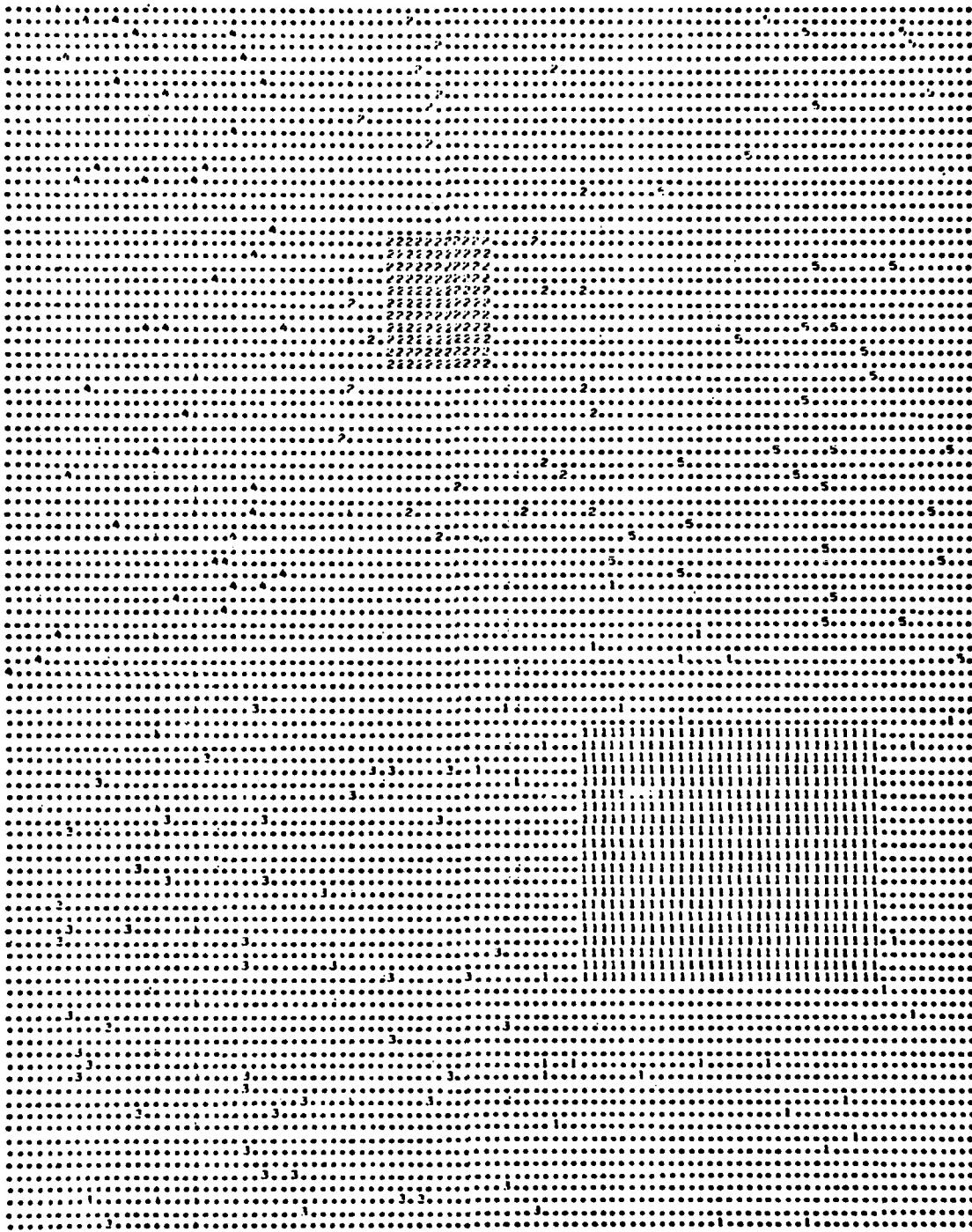


Figure 2.14: x-y Clustering, Low Background Level; NC=5

## MODELING OF SHADOWS IN RADAR CLUTTER

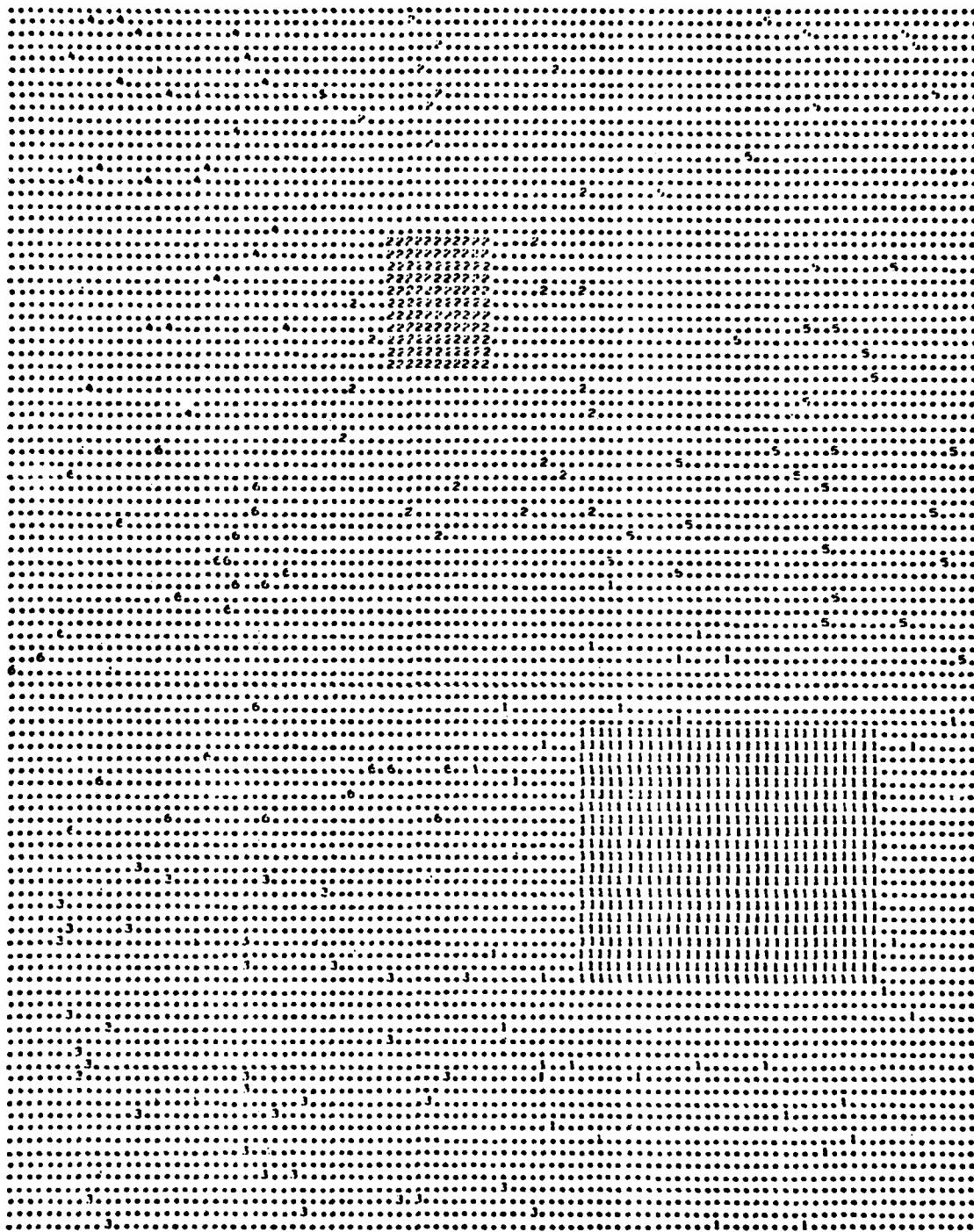


Figure 2.15: x-y Clustering, Low Background Level; NC=6

## MODELING OF SHADOWS IN RADAR CLUTTER

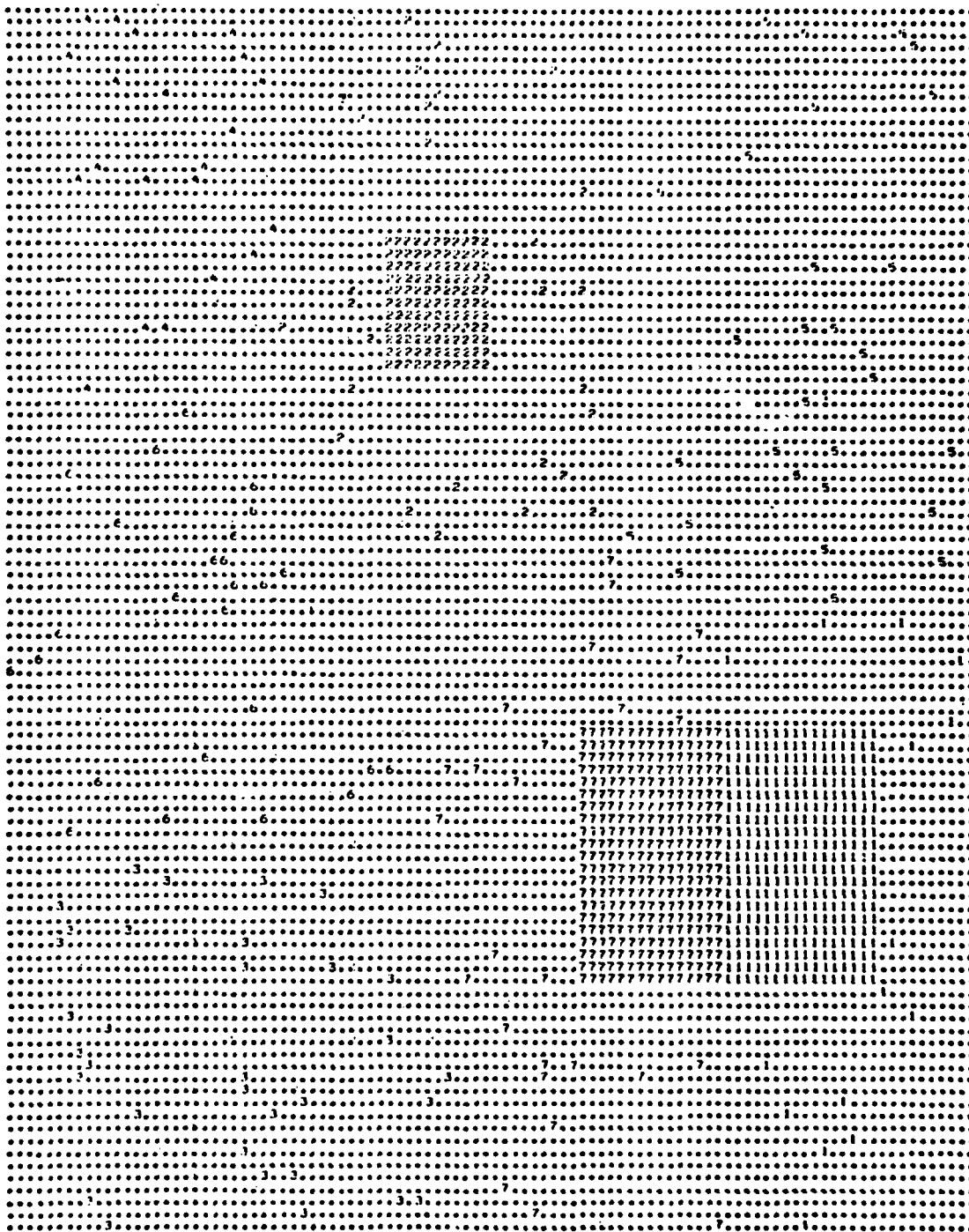


Figure 2.16: x-y Clustering, Low Background Level; NC=7

## MODELING OF SHADOWS IN RADAR CLUTTER

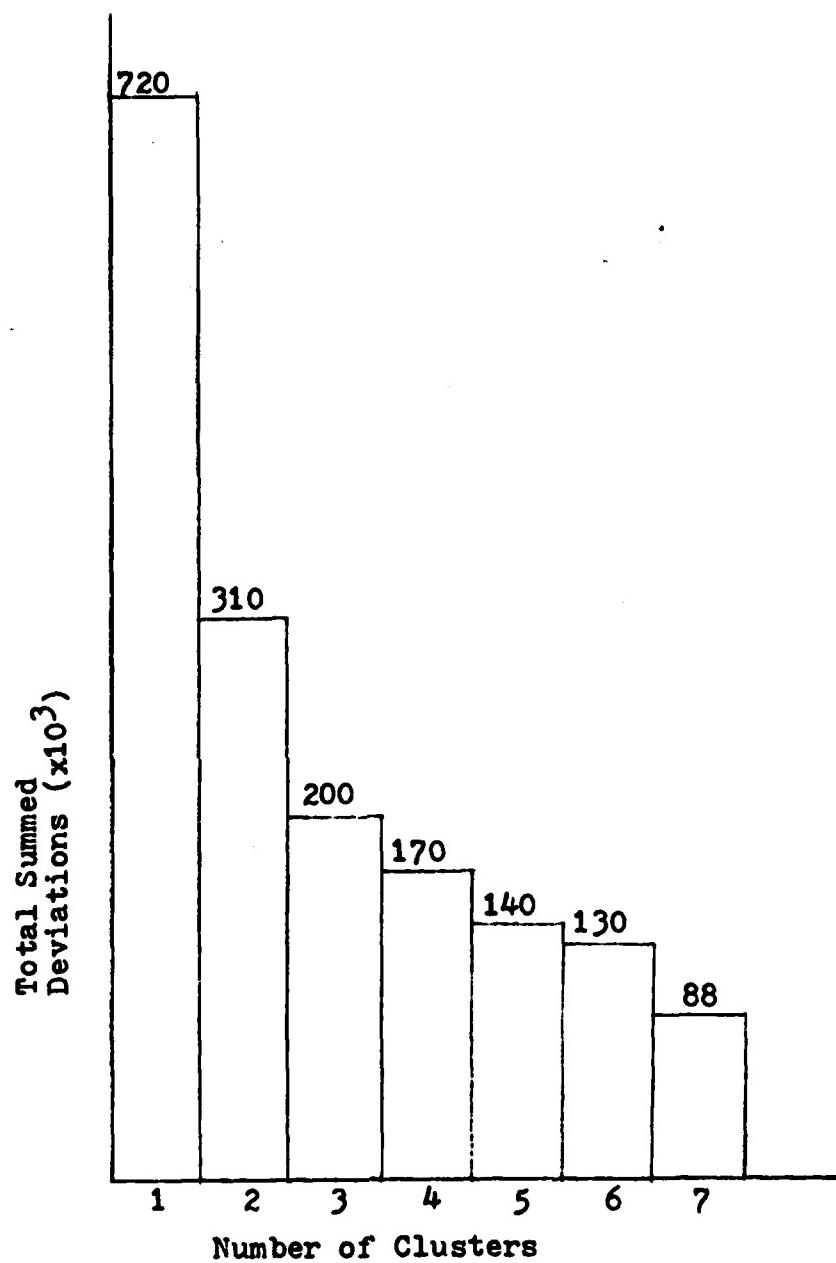


Figure 2.17: Total Summed Deviations

## MODELING OF SHADOWS IN RADAR CLUTTER

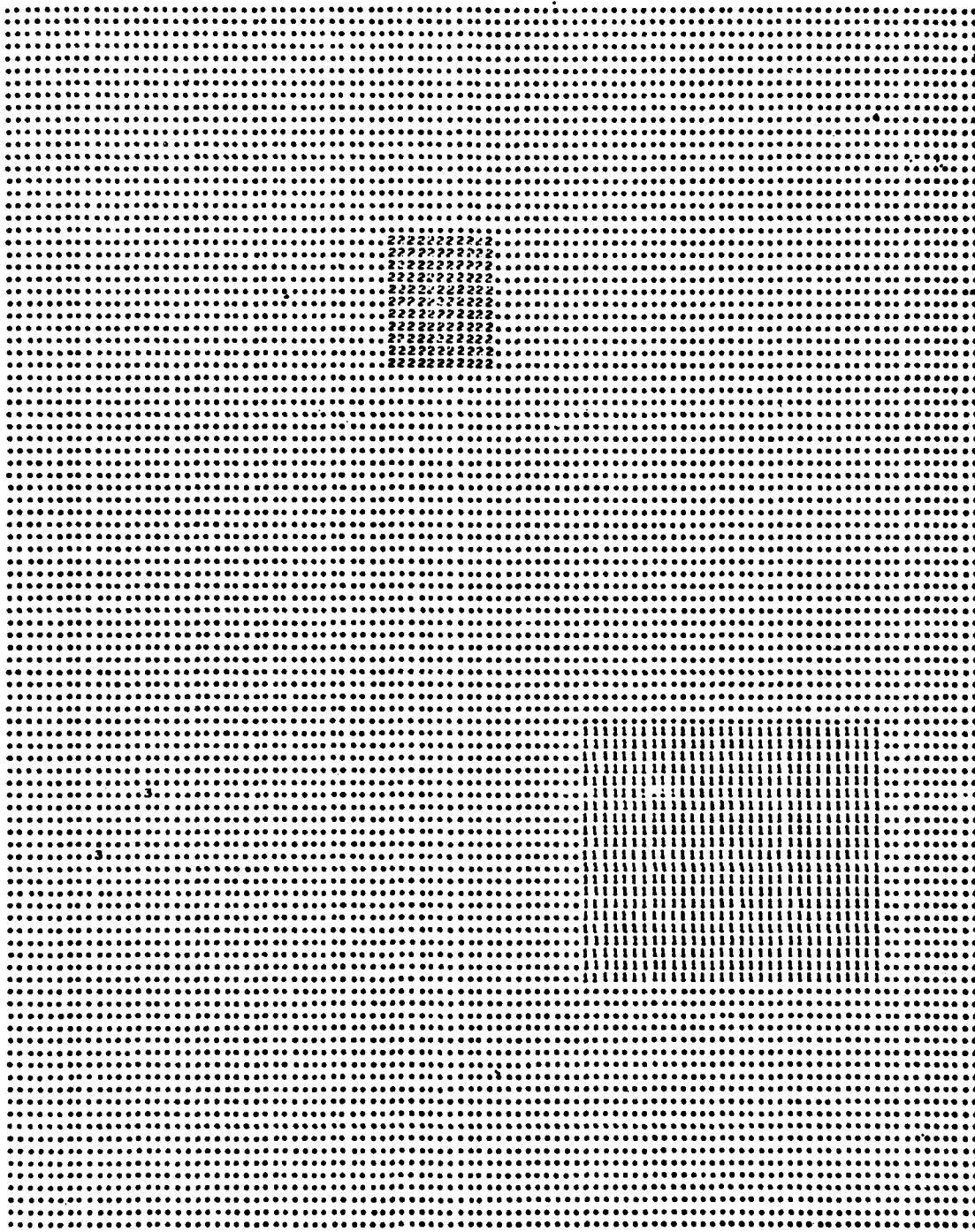


Figure 2.18: Test Scene With Outliers

MODELING OF SHADOWS IN RADAR CLUTTER

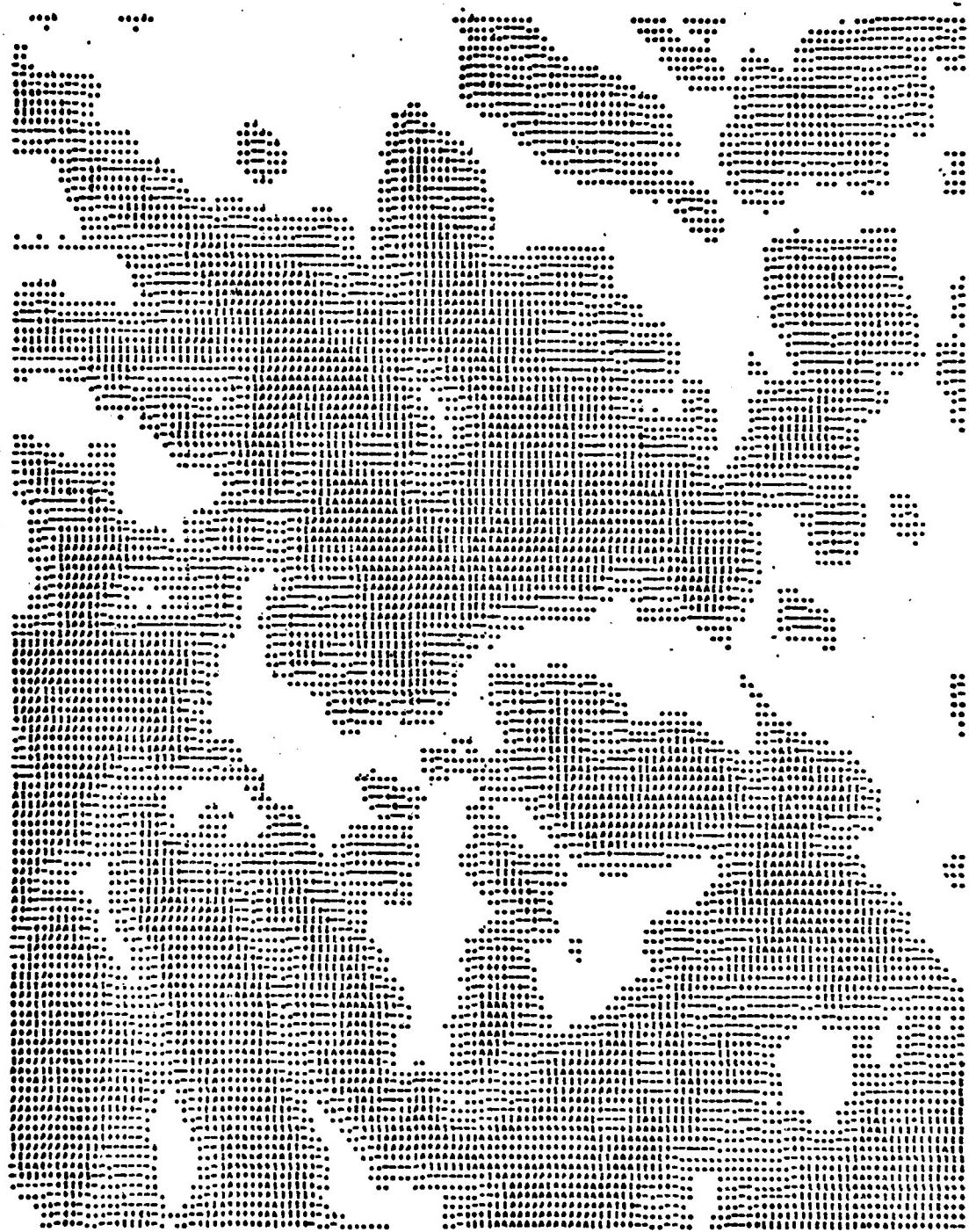


Figure 2.19: Radar Data

MODELING OF SHADOWS IN RADAR CLUTTER



Figure 2.20: Low Intensity Regions

## MODELING OF SHADOWS IN RADAR CLUTTER



Figure 2.21: x-y Clustering of Low Intensity Regions

# MODELING OF SHADOWS IN RADAR CLUTTER

168	2	10.	46.	2.0	152	2	11.	47.	0.0	167	6	10.	46.	0.0	164	9	32
170	3	13.	46.	0.0	172	3	14.	46.	0.0	173	6	12.	46.	0.0	172	6	37
171	3	15.	46.	0.0	172	3	16.	46.	0.0	173	6	13.	46.	0.0	172	6	37
172	3	17.	46.	0.0	173	3	18.	46.	0.0	174	6	14.	46.	0.0	173	6	37
173	0	18.	46.	0.0	174	0	19.	46.	0.0	175	6	15.	46.	0.0	174	6	37
174	1	19.	46.	0.0	175	1	20.	46.	0.0	176	6	16.	46.	0.0	175	6	37
175	1	21.	46.	0.0	176	1	22.	46.	0.0	177	6	17.	46.	0.0	176	6	37
176	1	22.	46.	0.0	177	1	23.	46.	0.0	178	6	18.	46.	0.0	177	6	37
177	1	23.	46.	0.0	178	1	24.	46.	0.0	179	6	19.	46.	0.0	178	6	37
178	0	24.	46.	0.0	179	0	25.	46.	0.0	180	6	20.	46.	0.0	179	6	37
179	0	25.	46.	0.0	180	0	26.	46.	0.0	181	6	21.	46.	0.0	180	6	37
180	0	26.	46.	0.0	181	0	27.	46.	0.0	182	6	22.	46.	0.0	181	6	37
181	0	27.	46.	0.0	182	0	28.	46.	0.0	183	6	23.	46.	0.0	182	6	37
182	0	28.	46.	0.0	183	0	29.	46.	0.0	184	6	24.	46.	0.0	183	6	37
183	0	29.	46.	0.0	184	0	30.	46.	0.0	185	6	25.	46.	0.0	184	6	37
184	0	30.	46.	0.0	185	0	31.	46.	0.0	186	6	26.	46.	0.0	185	6	37
185	0	31.	46.	0.0	186	0	32.	46.	0.0	187	6	27.	46.	0.0	186	6	37
186	0	32.	46.	0.0	187	0	33.	46.	0.0	188	6	28.	46.	0.0	187	6	37
187	0	33.	46.	0.0	188	0	34.	46.	0.0	189	6	29.	46.	0.0	188	6	37
188	0	34.	46.	0.0	189	0	35.	46.	0.0	190	6	30.	46.	0.0	189	6	37
189	0	35.	46.	0.0	190	0	36.	46.	0.0	191	6	31.	46.	0.0	190	6	37
190	0	36.	46.	0.0	191	0	37.	46.	0.0	192	6	32.	46.	0.0	191	6	37
191	0	37.	46.	0.0	192	0	38.	46.	0.0	193	6	33.	46.	0.0	192	6	37
192	0	38.	46.	0.0	193	0	39.	46.	0.0	194	6	34.	46.	0.0	193	6	37
193	0	39.	46.	0.0	194	0	40.	46.	0.0	195	6	35.	46.	0.0	194	6	37
194	0	40.	46.	0.0	195	0	41.	46.	0.0	196	6	36.	46.	0.0	195	6	37
195	0	41.	46.	0.0	196	0	42.	46.	0.0	197	6	37.	46.	0.0	196	6	37
196	0	42.	46.	0.0	197	0	43.	46.	0.0	198	6	38.	46.	0.0	197	6	37
197	0	43.	46.	0.0	198	0	44.	46.	0.0	199	6	39.	46.	0.0	198	6	37
198	0	44.	46.	0.0	199	0	45.	46.	0.0	200	6	40.	46.	0.0	199	6	37
199	0	45.	46.	0.0	200	0	46.	46.	0.0	201	6	41.	46.	0.0	200	6	37
200	0	46.	46.	0.0	201	0	47.	46.	0.0	202	6	42.	46.	0.0	201	6	37
201	0	47.	46.	0.0	202	0	48.	46.	0.0	203	6	43.	46.	0.0	202	6	37
202	0	48.	46.	0.0	203	0	49.	46.	0.0	204	6	44.	46.	0.0	203	6	37
203	0	49.	46.	0.0	204	0	50.	46.	0.0	205	6	45.	46.	0.0	204	6	37
204	0	50.	46.	0.0	205	0	51.	46.	0.0	206	6	46.	46.	0.0	205	6	37
205	0	51.	46.	0.0	206	0	52.	46.	0.0	207	6	47.	46.	0.0	206	6	37
206	0	52.	46.	0.0	207	0	53.	46.	0.0	208	6	48.	46.	0.0	207	6	37
207	0	53.	46.	0.0	208	0	54.	46.	0.0	209	6	49.	46.	0.0	208	6	37
208	0	54.	46.	0.0	209	0	55.	46.	0.0	210	6	50.	46.	0.0	209	6	37
209	0	55.	46.	0.0	210	0	56.	46.	0.0	211	6	51.	46.	0.0	210	6	37
210	0	56.	46.	0.0	211	0	57.	46.	0.0	212	6	52.	46.	0.0	211	6	37
211	0	57.	46.	0.0	212	0	58.	46.	0.0	213	6	53.	46.	0.0	212	6	37
212	0	58.	46.	0.0	213	0	59.	46.	0.0	214	6	54.	46.	0.0	213	6	37
213	0	59.	46.	0.0	214	0	60.	46.	0.0	215	6	55.	46.	0.0	214	6	37
214	0	60.	46.	0.0	215	0	61.	46.	0.0	216	6	56.	46.	0.0	215	6	37

### X-Y RAYLEIGH DATA

I-4 DIST. WHEN I GET THROUGH WITH IT, YOUR DATA IS GONNA BE WAYYY OUT.

NIN 30 CREATED IN INPU NPAT 218 NCST 0 NVAR 2 NCAT 1 IFER 0

ROUT 30

NTYPE 0 MAHALANOBIS DISTANCE OF CPDFN 1  
(IE, EUCLIDEAN DISTANCE...)

### X-Y RAYLEIGH DATA

I-4 TREE. I REALLY PUT YOUR DATA OUT ON A LIMB.

NIN 30 CREATED IN DIST NPAT 218 NCST 0 NVAR DMAX 1.202E 02 NCAT 1 IFER 0

VENT 1

NET 1 PRUNING PARAMETERS WILL BE READ FROH CARDS.

***** NODE *****		***** NEIGHBORS *****						***** DISTANCE *****		
1	0	2.	9.	0.0	2	0	2.	12.	0.0	3.000E 00
2	0	2.	12.	0.0	5	0	3.	14.	0.0	2.236E 00
3	0	2.	42.	0.0	9	0	4.	45.	0.0	3.600E 00
4	0	2.	79.	0.0	7	0	3.	83.	0.0	4.123E 00
5	0	3.	14.	0.0	15	0	7.	12.	0.0	4.672E 00
6	0	3.	24.	0.0	12	0	5.	25.	0.0	2.236E 00
7	0	3.	83.	0.0	21	0	9.	84.	0.0	6.093E 02
8	0	3.	93.	0.0	6	0	2.	74.	0.0	4.123E 00
9	0	3.	93.	0.0	8	0	3.	93.	0.0	1.000E 01
10	0	4.	83.	0.0	7	0	3.	83.	0.0	1.000E 01
11	0	4.	84.	0.0	10	0	4.	84.	0.0	1.414E 00

Figure 3.1: TREE Example Run

## MODELING OF SHADOWS IN RADAR CLUTTER

THE FILE WAS READ CLUSTERED SEARCHING FOR A CENTER OF 2.  
 INCONSISTENT CODES HAVE BEEN REMOVED BY A FACTOR OF 2.0000E 00  
 AND A SHIFT OF -0.0 STANDARD DEVIATIONS.

9 CLUSTERS IDENTIFIED			
CLUSTER	1	2	3
1	2	2	2
2	0	2	2
3	0	2	2
4	0	2	2
5	0	2	2
6	0	2	2
7	0	2	2
8	0	2	2
9	0	2	2
10	0	2	2
11	0	2	2
12	0	2	2
13	0	2	2
14	0	2	2
15	0	2	2
16	0	2	2
17	0	2	2
18	0	2	2
19	0	2	2
20	0	2	2
21	0	2	2
22	0	2	2
23	0	2	2
24	0	2	2
25	0	2	2
26	0	2	2
27	0	2	2
28	0	2	2
29	0	2	2
30	0	2	2
31	0	2	2
32	0	2	2
33	0	2	2
34	0	2	2
35	0	2	2
36	0	2	2
37	0	2	2
38	0	2	2
39	0	2	2
40	0	2	2
41	0	2	2
42	0	2	2
43	0	2	2
44	0	2	2
45	0	2	2
46	0	2	2
47	0	2	2
48	0	2	2
49	0	2	2
50	0	2	2
51	0	2	2
52	0	2	2
53	0	2	2
54	0	2	2
55	0	2	2
56	0	2	2
57	0	2	2
58	0	2	2
59	0	2	2
60	0	2	2
61	0	2	2
62	0	2	2
63	0	2	2
64	0	2	2
65	0	2	2
66	0	2	2
67	0	2	2
68	0	2	2
69	0	2	2
70	0	2	2
71	0	2	2
72	0	2	2
73	0	2	2
74	0	2	2
75	0	2	2
76	0	2	2
77	0	2	2
78	0	2	2
79	0	2	2
80	0	2	2
81	0	2	2
82	0	2	2
83	0	2	2
84	0	2	2
85	0	2	2
86	0	2	2
87	0	2	2
88	0	2	2
89	0	2	2
90	0	2	2
91	0	2	2
92	0	2	2
93	0	2	2
94	0	2	2
95	0	2	2
96	0	2	2
97	0	2	2
98	0	2	2
99	0	2	2
100	0	2	2
101	0	2	2
102	0	2	2
103	0	2	2
104	0	2	2
105	0	2	2
106	0	2	2
107	0	2	2
108	0	2	2
109	0	2	2
110	0	2	2
111	0	2	2
112	0	2	2
113	0	2	2
114	0	2	2
115	0	2	2
116	0	2	2
117	0	2	2
118	0	2	2
119	0	2	2
120	0	2	2
121	0	2	2
122	0	2	2
123	0	2	2
124	0	2	2
125	0	2	2
126	0	2	2
127	0	2	2
128	0	2	2
129	0	2	2
130	0	2	2
131	0	2	2
132	0	2	2
133	0	2	2
134	0	2	2
135	0	2	2
136	0	2	2
137	0	2	2
138	0	2	2
139	0	2	2
140	0	2	2
141	0	2	2
142	0	2	2
143	0	2	2
144	0	2	2
145	0	2	2
146	0	2	2
147	0	2	2
148	0	2	2
149	0	2	2
150	0	2	2
151	0	2	2
152	0	2	2
153	0	2	2
154	0	2	2
155	0	2	2
156	0	2	2
157	0	2	2
158	0	2	2
159	0	2	2
160	0	2	2
161	0	2	2
162	0	2	2
163	0	2	2
164	0	2	2
165	0	2	2
166	0	2	2
167	0	2	2
168	0	2	2
169	0	2	2
170	0	2	2
171	0	2	2
172	0	2	2
173	0	2	2
174	0	2	2
175	0	2	2
176	0	2	2
177	0	2	2
178	0	2	2
179	0	2	2
180	0	2	2
181	0	2	2
182	0	2	2
183	0	2	2
184	0	2	2
185	0	2	2
186	0	2	2
187	0	2	2
188	0	2	2
189	0	2	2
190	0	2	2
191	0	2	2
192	0	2	2
193	0	2	2
194	0	2	2
195	0	2	2
196	0	2	2
197	0	2	2
198	0	2	2
199	0	2	2
200	0	2	2
201	0	2	2
202	0	2	2
203	0	2	2
204	0	2	2
205	0	2	2
206	0	2	2
207	0	2	2
208	0	2	2
209	0	2	2
210	0	2	2
211	0	2	2
212	0	2	2
213	0	2	2
214	0	2	2
215	0	2	2
216	0	2	2
217	0	2	2
218	0	2	2
219	0	2	2
220	0	2	2
221	0	2	2
222	0	2	2
223	0	2	2
224	0	2	2
225	0	2	2
226	0	2	2
227	0	2	2
228	0	2	2
229	0	2	2
230	0	2	2
231	0	2	2
232	0	2	2
233	0	2	2
234	0	2	2
235	0	2	2
236	0	2	2
237	0	2	2
238	0	2	2
239	0	2	2
240	0	2	2
241	0	2	2
242	0	2	2
243	0	2	2
244	0	2	2
245	0	2	2
246	0	2	2
247	0	2	2
248	0	2	2
249	0	2	2
250	0	2	2
251	0	2	2
252	0	2	2
253	0	2	2
254	0	2	2
255	0	2	2
256	0	2	2
257	0	2	2
258	0	2	2
259	0	2	2
260	0	2	2
261	0	2	2
262	0	2	2
263	0	2	2
264	0	2	2
265	0	2	2
266	0	2	2
267	0	2	2
268	0	2	2
269	0	2	2
270	0	2	2
271	0	2	2
272	0	2	2
273	0	2	2
274	0	2	2
275	0	2	2
276	0	2	2
277	0	2	2
278	0	2	2
279	0	2	2
280	0	2	2
281	0	2	2
282	0	2	2
283	0	2	2
284	0	2	2
285	0	2	2
286	0	2	2
287	0	2	2
288	0	2	2
289	0	2	2
290	0	2	2
291	0	2	2
292	0	2	2
293	0	2	2
294	0	2	2
295	0	2	2
296	0	2	2
297	0	2	2
298	0	2	2
299	0	2	2
300	0	2	2
301	0	2	2
302	0	2	2
303	0	2	2
304	0	2	2
305	0	2	2
306	0	2	2
307	0	2	2
308	0	2	2
309	0	2	2
310	0	2	2
311	0	2	2
312	0	2	2
313	0	2	2
314	0	2	2
315	0	2	2
316	0	2	2
317	0	2	2
318	0	2	2
319	0	2	2
320	0	2	2
321	0	2	2
322	0	2	2
323	0	2	2
324	0	2	2
325	0	2	2
326	0	2	2
327	0	2	2
328	0	2	2
329	0	2	2
330	0	2	2
331	0	2	2
332	0	2	2
333	0	2	2
334	0	2	2
335	0	2	2
336	0	2	2
337	0	2	2
338	0	2	2
339	0	2	2
340	0	2	2
341	0	2	2
342	0	2	2
343	0	2	2
344	0	2	2
345	0	2	2
346	0	2	2
347	0	2	2
348	0	2	2
349	0	2	2
350	0	2	2
351	0	2	2
352	0	2	2
353	0	2	2
354	0	2	2
355	0	2	2
356	0	2	2
357	0	2	2
358	0	2	2
359	0	2	2
360	0	2	2
361	0	2	2
362	0	2	2
363			

## MODELING OF SHADOWS IN RADAR CLUTTER

THE DATA HAS BEEN CLUSTERED GROUPING IN 7 CLUSTERS. THE INCORPORATED PATTERNS HAVE BEEN STRETCHED BY A FACTOR OF 1.000000. 00 AND A SHIFT OF 1.000000 STANDARD DEVIATIONS.

### 3.2 CLUSTERS DETECTED

CLUSTER	CONSISTS OF THE FOLLOWING	PATTERNS
1	0 26 9.	0.0
2	0 26 12.	0.0
3	0 30 14.	0.0
15	0 7. 12.	0.0
17	0 8. 17.	0.0
CLUSTER 2 CONSISTS OF THE FOLLOWING		4 PATTERNS
27	0 14. 21.	0.0
30	0 15. 20.	0.0
23	0 11. 24.	0.0
29	0 15. 15.	0.0
CLUSTER 3 CONSISTS OF THE FOLLOWING		2 PATTERNS
26	0 14. 15.	0.0
28	0 15. 8.	0.0
CLUSTER 4 CONSISTS OF THE FOLLOWING		3 PATTERNS
16	0 7. 27.	0.0
12	0 5. 25.	0.0
6	0 3. 24.	0.0
CLUSTER 5 CONSISTS OF THE FOLLOWING		5 PATTERNS
70	0 23. 22.	0.0
119	0 27. 17.	0.0
66	0 21. 26.	0.0
118	0 27. 15.	0.0
33	0 19. 26.	0.0
CLUSTER 6 CONSISTS OF THE FOLLOWING		1 PATTERNS
120	0 27. 29.	0.0
CLUSTER 7 CONSISTS OF THE FOLLOWING		122 PATTERNS
95	0 25. 36.	0.0
133	0 25. 32.	0.0
134	0 25. 40.	0.0
166	0 25. 40.	0.0
135	0 25. 41.	0.0
121	0 27. 40.	0.0
157	0 30. 40.	0.0
147	0 25. 41.	0.0
122	0 27. 41.	0.0
197	0 26. 40.	0.0
158	0 33. 41.	0.0
109	0 26. 41.	0.0
96	0 25. 42.	0.0
159	0 30. 42.	0.0
97	0 25. 41.	0.0
82	0 26. 40.	0.0
160	0 33. 43.	0.0
148	0 29. 42.	0.0
82	0 24. 41.	0.0
71	0 23. 40.	0.0
161	0 30. 41.	0.0
199	0 24. 43.	0.0
136	0 28. 42.	0.0

Figure 3.3: Better Clustering

## MODELING OF SHADOWS IN RADAR CLUTTER

THE TREE HAS BEEN CLUSTERED SEARCHING TO A DEPTH OF 1.  
INCONSISTENT EDGES HAVE BEEN DETERMINED BY A FACTOR OF 1.0000E 00  
AND A SWEEP OF 5.0000E-01 STANDARD DEVIATIONS.

### 17 CLUSTERS DETECTED

CLUSTER 1 CONSISTS OF THE FOLLOWING 3 PATTERNS.

1	0	2.	9.	0.0
2	0	3.	12.	0.0
5	0	3.	14.	0.0

CLUSTER 2 CONSISTS OF THE FOLLOWING 2 PATTERNS.

15	0	7.	12.	0.0
17	0	8.	17.	0.0

CLUSTER 3 CONSISTS OF THE FOLLOWING 4 PATTERNS.

27	0	14.	21.	0.0
33	0	15.	20.	0.0
23	0	11.	24.	0.0
29	0	15.	15.	0.0

CLUSTER 4 CONSISTS OF THE FOLLOWING 2 PATTERNS.

26	0	14.	19.	0.0
28	0	15.	8.	0.0

CLUSTER 5 CONSISTS OF THE FOLLOWING 3 PATTERNS.

10	0	7.	27.	0.0
12	0	5.	25.	0.0
6	0	3.	24.	0.0

CLUSTER 6 CONSISTS OF THE FOLLOWING 5 PATTERNS.

70	0	23.	22.	0.0
119	0	27.	17.	0.0
46	0	21.	26.	0.0
118	0	27.	15.	0.0
22	0	19.	28.	0.0

CLUSTER 7 CONSISTS OF THE FOLLOWING 1 PATTERNS.

123	0	27.	29.	0.0
-----	---	-----	-----	-----

CLUSTER 8 CONSISTS OF THE FOLLOWING 123 PATTERNS.

65	0	25.	36.	0.0
133	0	28.	31.	0.0
134	0	28.	40.	0.0
136	0	34.	40.	0.0
125	0	28.	41.	0.0
121	0	27.	40.	0.0
127	0	33.	40.	0.0
137	0	24.	41.	0.0
141	0	27.	41.	0.0
107	0	26.	40.	0.0
108	0	33.	41.	0.0
106	0	26.	41.	0.0
66	0	26.	40.	0.0
109	0	33.	47.	0.0
27	0	25.	41.	0.0
67	0	34.	40.	0.0
163	0	30.	43.	0.0
144	0	24.	42.	0.0
111	0	24.	41.	0.0
141	0	31.	32.	0.0
141	0	30.	44.	0.0

Figure 3.4: Best Clustering Found

MODELING OF SHADOWS IN RADAR CLUTTER

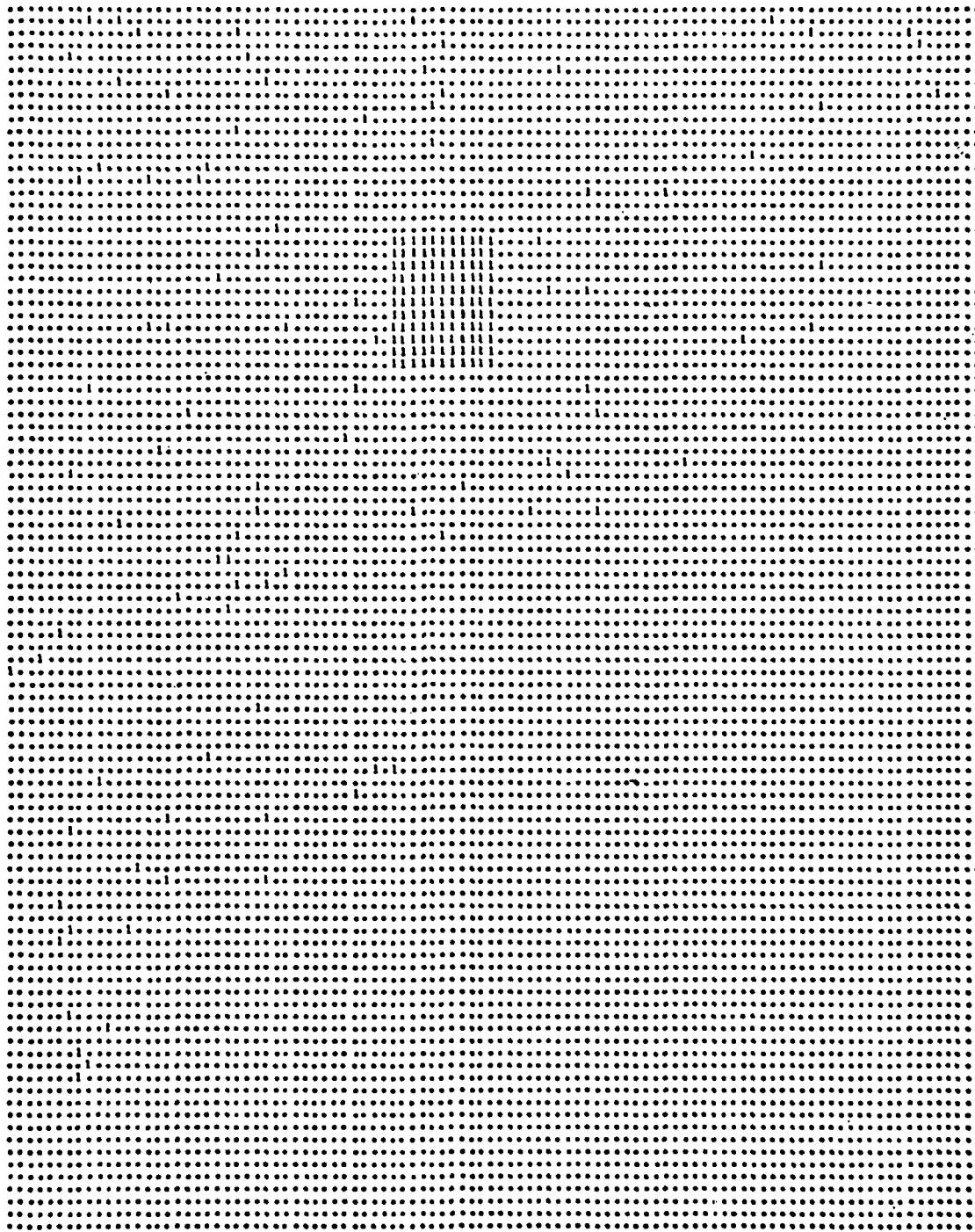


Figure 3.5: Data Set for Tree Example

MODELING OF SHADOWS IN RADAR CLUTTER

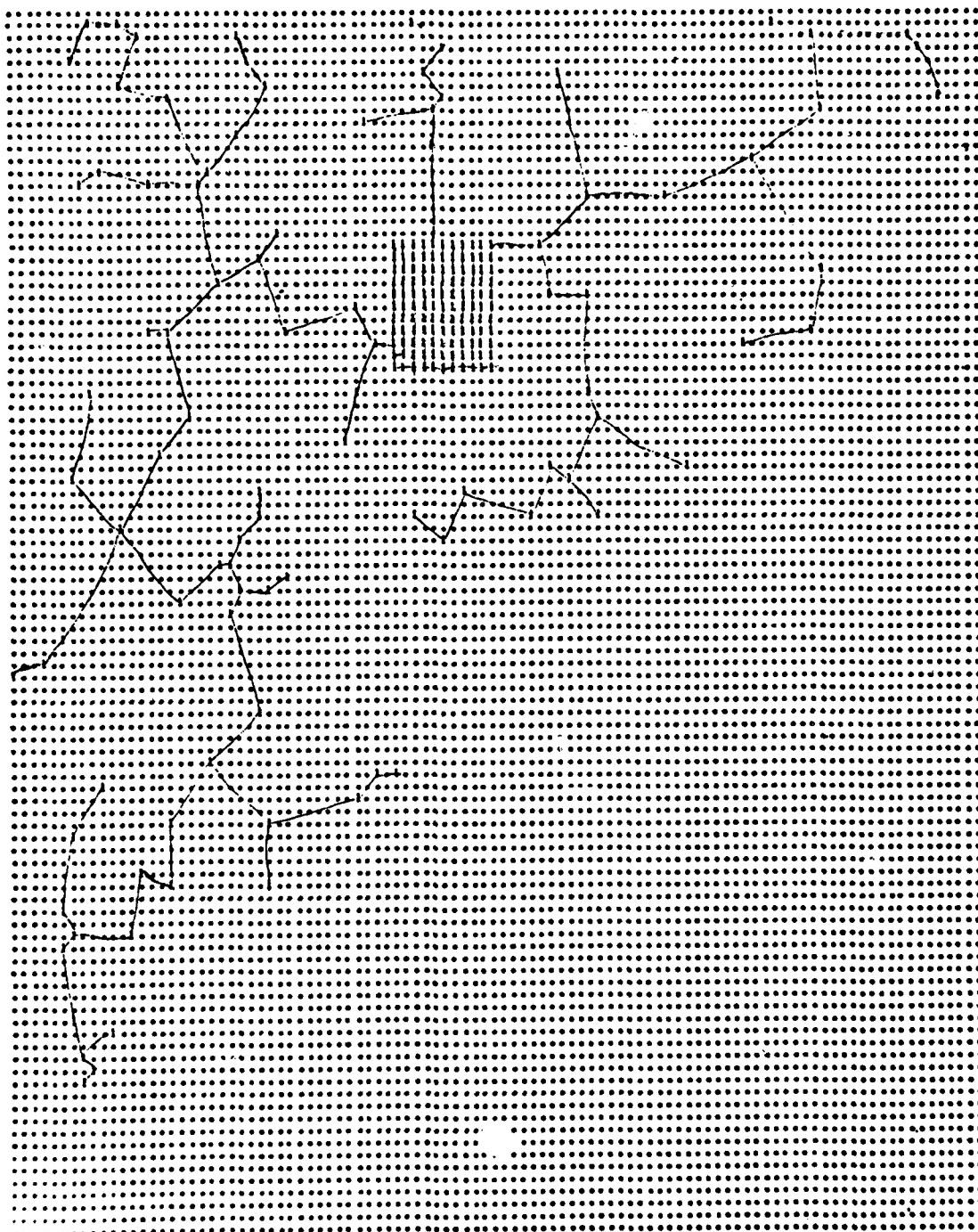


Figure 3.6: Minimal Spanning Tree

MODELING OF SHADOWS IN RADAR CLUTTER

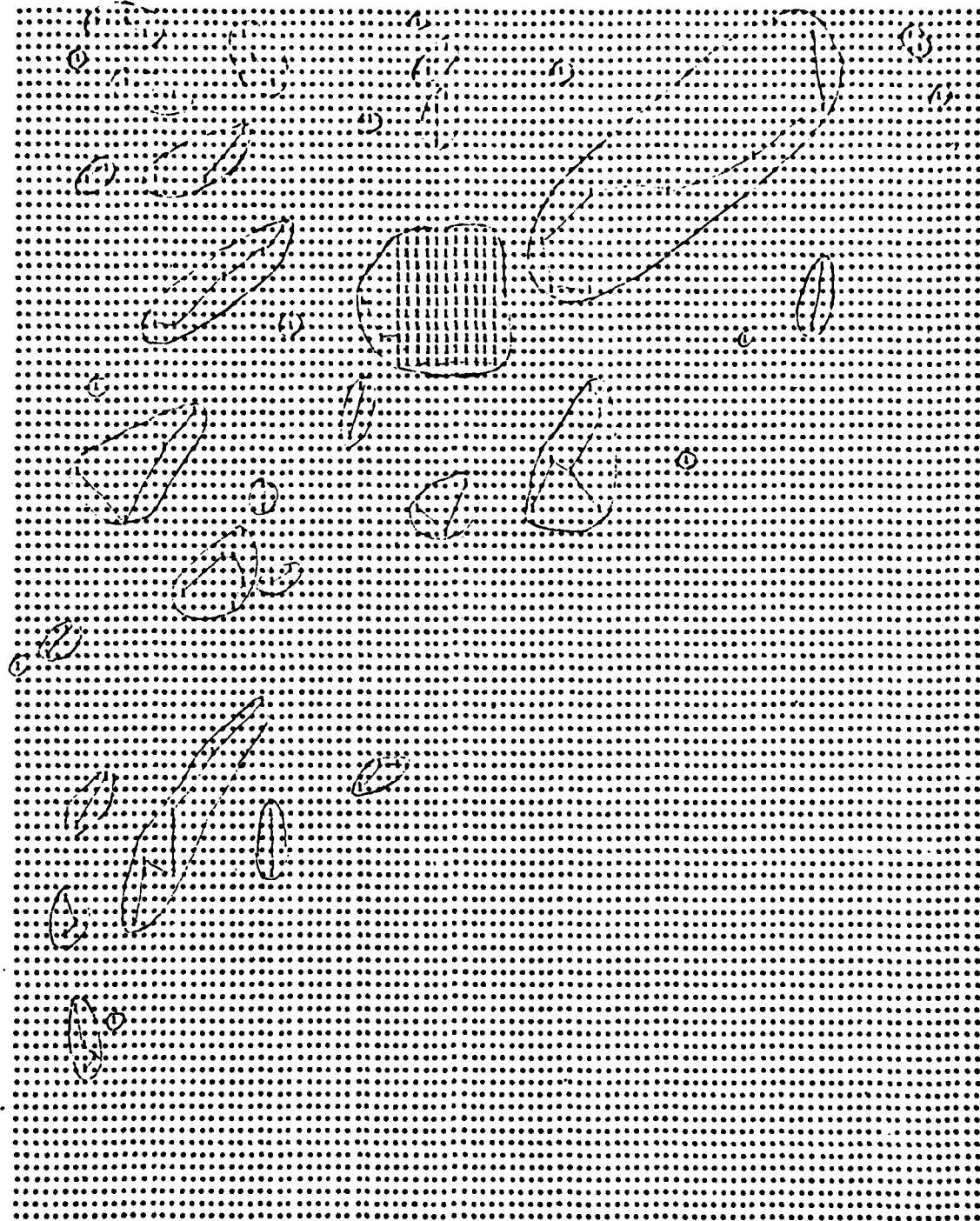


Figure 3.7: Best Clustering Found

MODELING OF SHADOWS IN RADAR CLUTTER



Figure 4.1: Point-to-Point Local Variance

MODELING OF SHADOWS IN RADAR CLUTTER

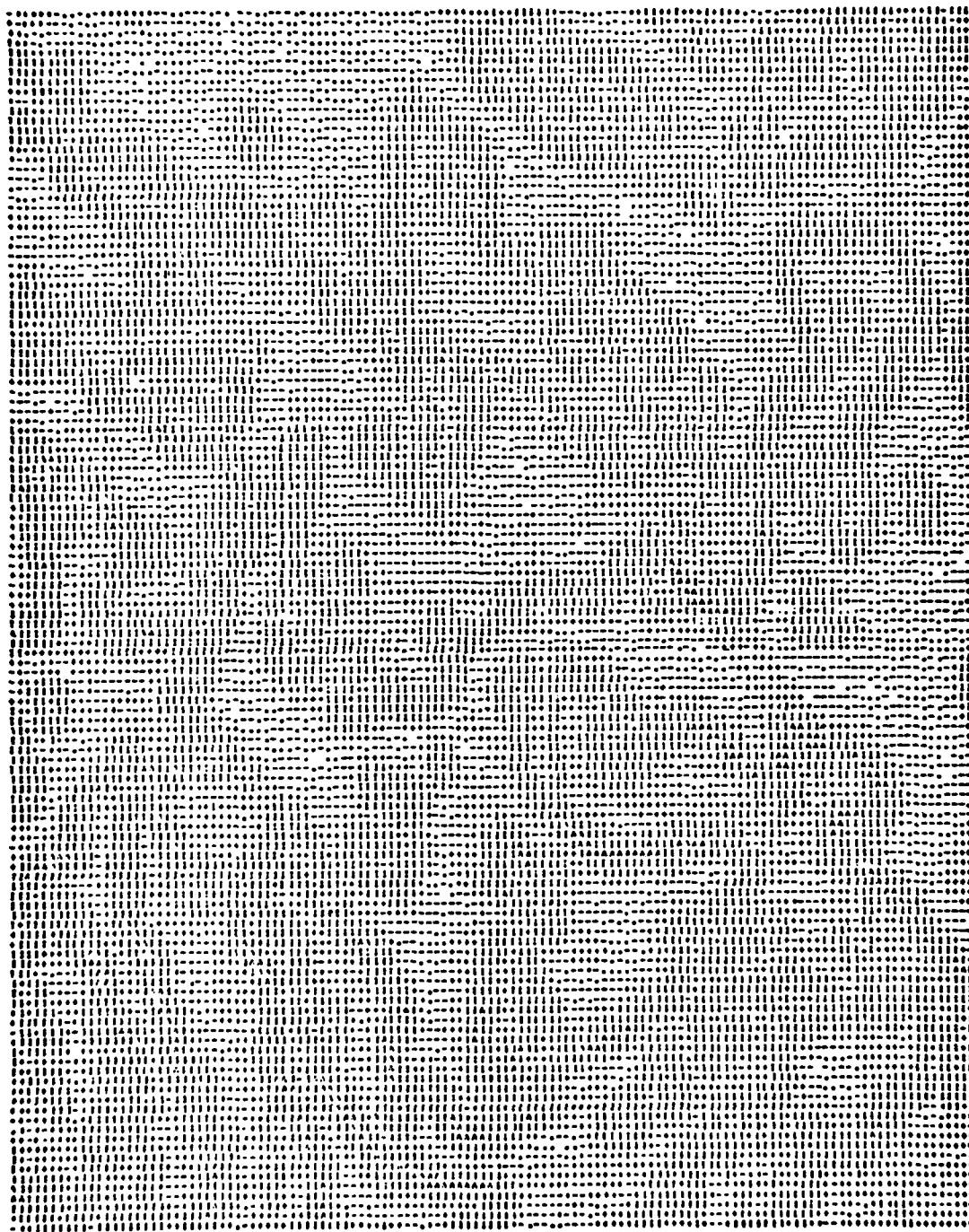


Figure 4.2: Point-to-Point Local Gradient

## MODELING OF SHADOWS IN RADAR CLUTTER



Figure 4.3: Location of Radar in Upper Quadrant

MODELING OF SHADOWS IN RADAR CLUTTER



Figure 4.4: Location of Radar in Lower Quadrant

MODELING OF SHADOWS IN RADAR CLUTTER

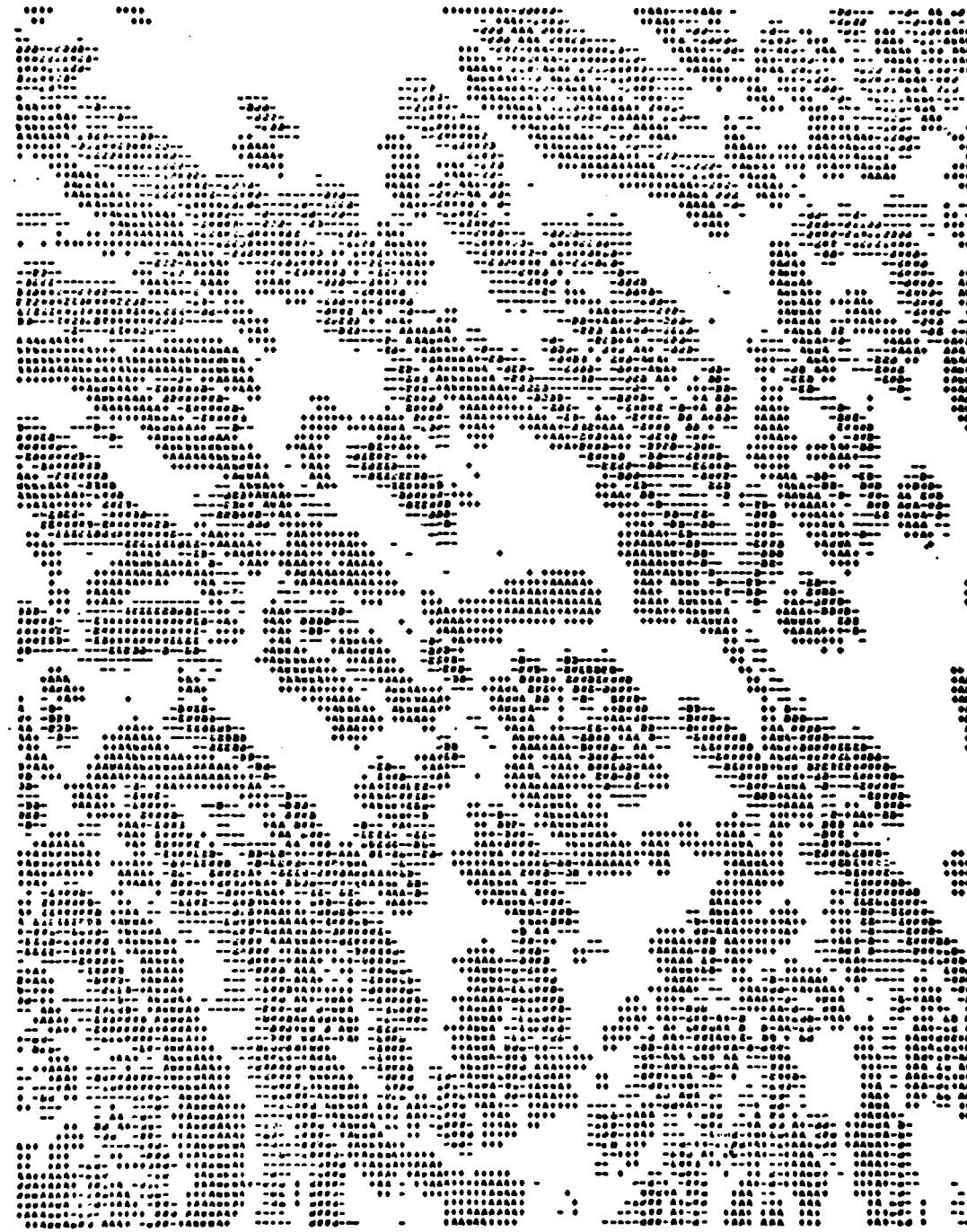


Figure 4.5: Radial Derivative, Upper Quadrant

MODELING OF SHADOWS IN RADAR CLUTTER

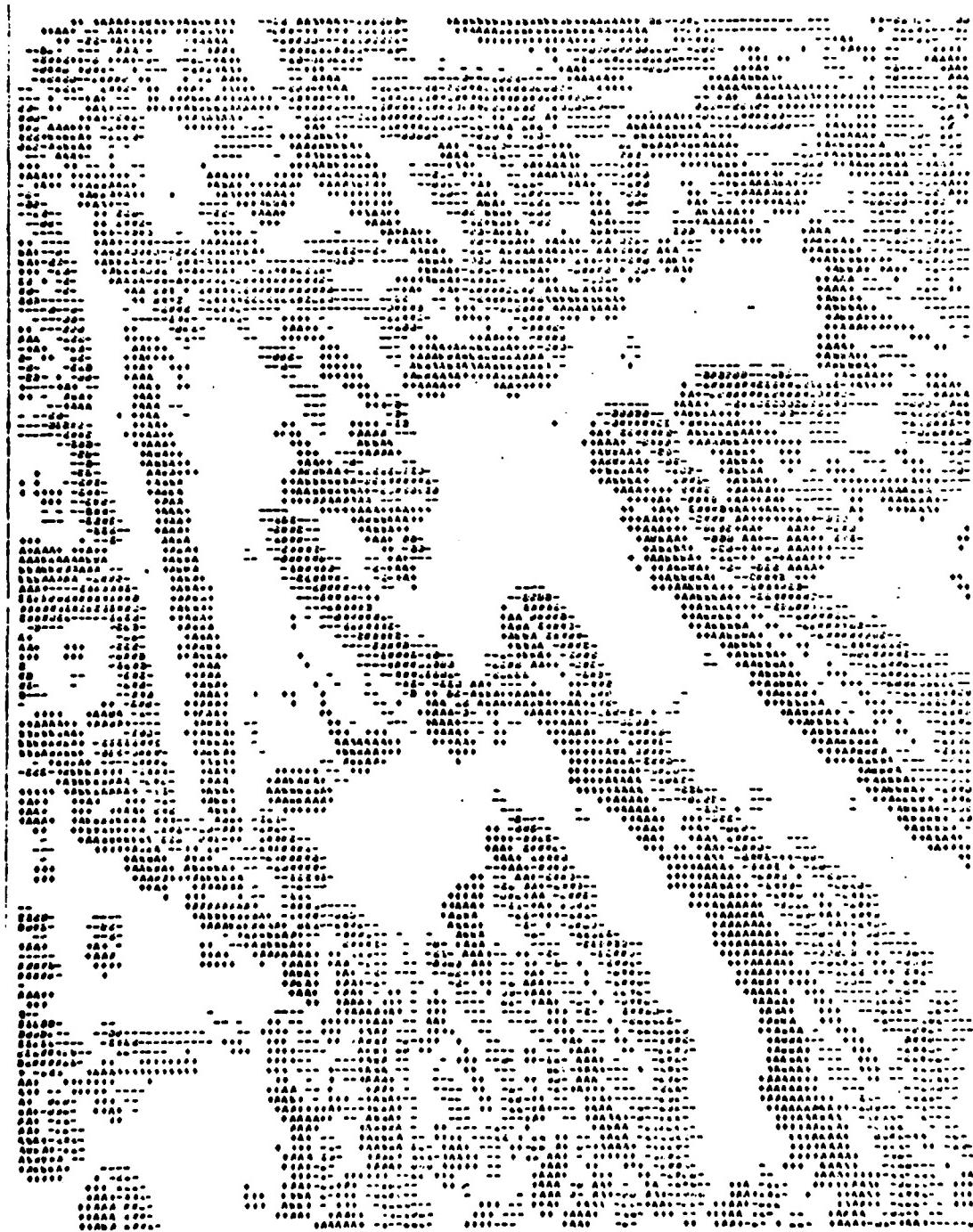


Figure 4.6: Radial Derivative, Lower Quadrant

MODELING OF SHADOWS IN RADAR CLUTTER

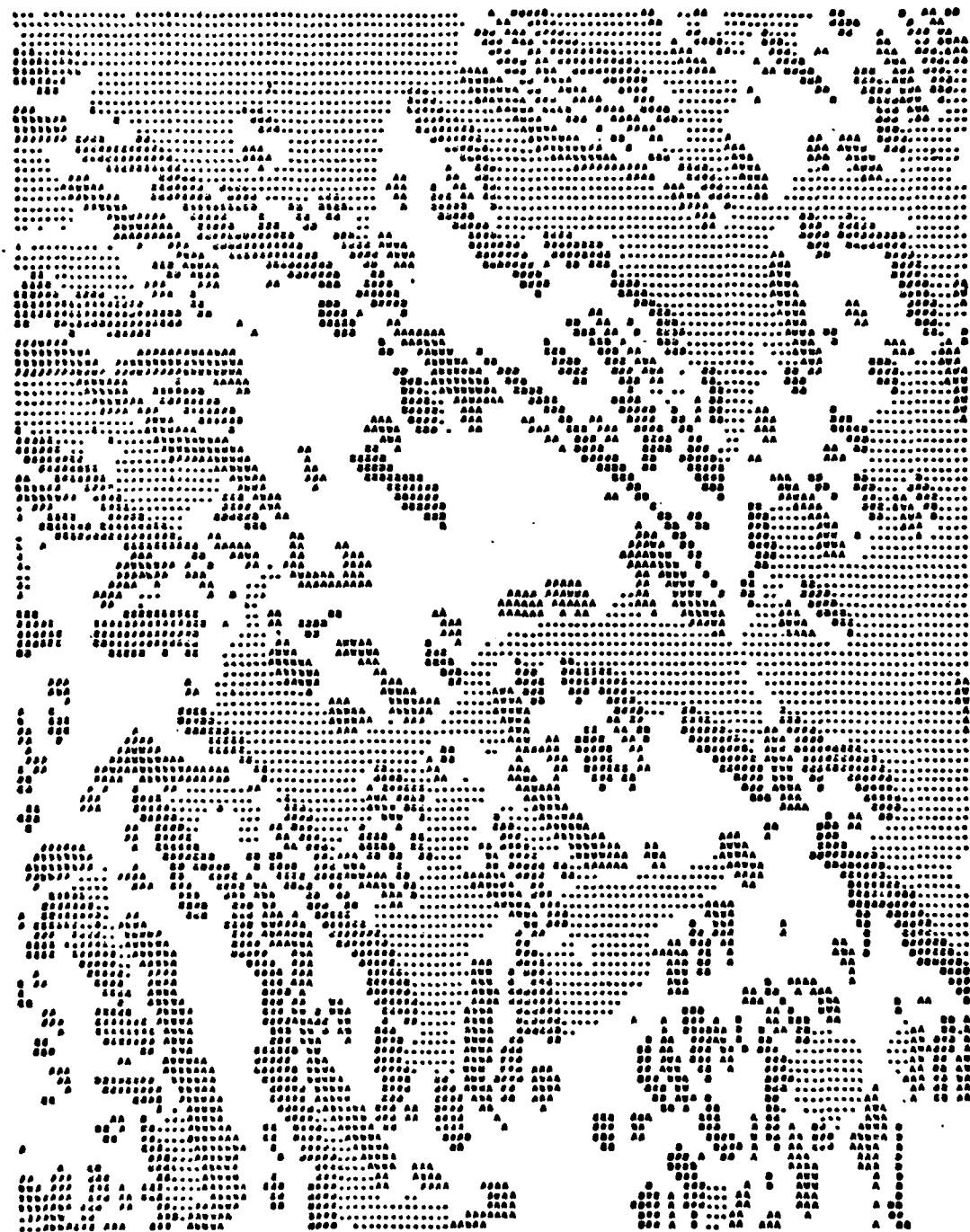


Figure 4.7: Radial Derivative and Shadows

## MODELING OF SHADOWS IN RADAR CLUTTER

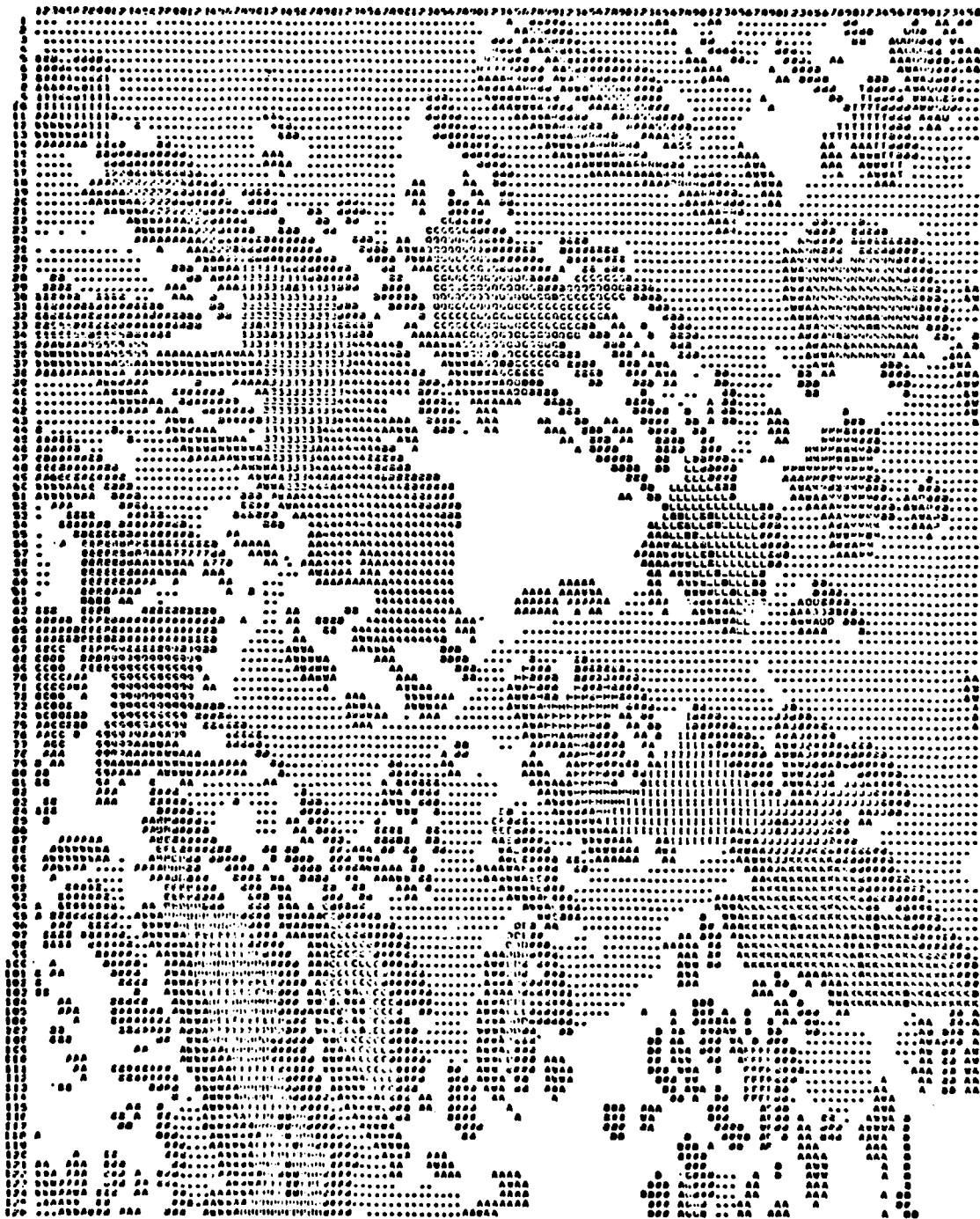


Figure 4.8: Precursors of Shadows

## MODELING OF SHADOWS IN RADAR CLUTTER

THIS SUBROUTINE READS PARAMETERS, COMPUTES STORAGE AND CALL'S MAJOR  
 PROGRAM SEGMENT, WHICH IS A NON-HIERARCHICAL CLUSTERING JOB USING  
 ONE OF THE METHODS PROGRAMMED AS A VERSION OF SUBROUTINE "KMEAN".  
 EVERY JOB REQUIRES THREE USER SUPPLIED DECK SEGMENTS.  
 1. PROGRAM "KMEAN" PERFORMS THE FOLLOWING TASKS:  
     A. ASSIGNING INPUT/OUTPUT UNITS.  
     B. ESTABLISHES THE DIMENSION OF THE "X" ARRAY AND SETS THIS  
         DIMENSION TO "NUNITS".  
     C. CALLS SUBROUTINE "EXEC2" (FORMERLY "EXEC")  
  
 2. SUBROUTINE "USER" IS EMPLOYED TO READ THE COMPLETE SET OF SCORES  
 ON THE VARIABLES FOR ONE DATA UNIT.  
  
 3. FUNCTION "DIST" COMPUTES THE DISTANCE BETWEEN TWO DATA UNITS OR  
 BETWEEN A DATA UNIT AND A CLUSTER CENTER. THE USER CAN SPECIFY  
 ANY DESIRED DISTANCE FUNCTION AND WEIGHT THE VARIABLES IN ANY  
 MANNER.  
  
 NOTE THAT SCALING AND TRANSFORMATION OF VARIABLES CAN BE  
 ACCOMPLISHED EITHER IN SUBROUTINE "USER" OR IN SUBROUTINE "DIST".  
  
 \*\*\*\*\*  
  
 NOTE: MODIFIED SUBROUTINE "EXEC" IS CALLED "EXEC2", AND CALLS  
 "RSLT2". "KMEAN" HAS NOT BEEN ALTERED.  
  
 \*\*\*\*\*  
  
 INPUT SPECIFICATIONS  
 CARD 1 TITLE  
 CARD 2 PARAMETER CARD  
 COLS 1-9 N=NUMBER OF ENTITIES (DATA UNITS)  
 COLS 9-13 NV=NUMBER OF VARIABLES  
 COLS 13-15 NC=NUMBER OF CLUSTERS  
 COLS 15-20 NTIN=INPUT UNIT FOR THE DATA SET  
 NTIN=NFS, CARD READER  
 NTIN=NFS, TAPE OR DISK FILE  
 COLS 21-22 NTOUT=OUTPUT UNIT FOR SAVING CLUSTER MEMBERSHIP LISTS  
 NTOUT=7, CARD PUNCH  
 NTOUT=L=0, DO NOT SAVE MEMBERSHIP LISTS  
 COLS 23-27 MINREL=TERMINATION PARAMETER. CLUSTERING ENDS WHEN A  
 CYCLE THROUGH THE DATA SET RESULTS IN "MINREL"  
 OR FEWER CHANGES IN CLUSTER MEMBERSHIPS  
 MINREL=L=0, ITERATE TO COMPLETE CONVERGENCE  
 COLS 28-29 IPART=INITIAL PARTITION PARAMETER  
 IPART=1, SEED POINTS ARE SELECTED FROM THE DATA JNETS.  
 READ THE SEQUENCE NUMBERS FOR THE CHOSEN DATA  
 UNITS FROM CARD(S) 3 IN 2010 FORMAT. IF THE  
 DATA SET IS NOT STORED IN CORE, THE LIST OF  
 SEQUENCE NUMBERS MUST BE IN ASCENDING ORDER  
 IPART=2, THE DATA UNITS ARE GROUPED INTO AN INITIAL  
 PARTITION IN THE INPUT SEQUENCE WITH THE  
 FIRST "NNUMBER(L)" IN CLUSTER 1, THE NEXT

C "NUVGR(?)": IN CLUSTER 2 ETC. READ THE  
 C "NUMGR" ARRAY FROM CARD(S) 3 IN 2040 FORMAT.  
 C IPART=3. THE SCOTE VECTORS FOR THE SEED POINTS ARE  
 C READ FROM CARD(S) 4 IN FORMAT "FMT" WHICH IS  
 C READ FROM CARD 3.  
 C COLS 30-31 METHOD=PARAMETER FOR CHOOSING THE ALGORITHM IN ONE  
 C VERSION OF SUBROUTINE "KMEAN".  
 C METHOD=1, JANCEY A.GORITHM  
 C METHOD=NE,1, FORGY ALGORITHM  
 C  
 C000 CARDS 3 AND 4 ARE READ IN SUBROUTINE "KMEAN" ACCORDING TO THE  
 C000 PROCEDURE SPECIFIED BY THE CHOSEN VALUE OF "IPART". NOTE THAT THE  
 C000 BASIC K-MEANS METHOD OF VACQUEMON SIMPLY USES THE FIRST "NC" DATA  
 C000 UNITS AS CLUSTER SEED POINTS AND THEREFORE IGNORES THE "(IPART)"  
 C000 PARAMETER.

STORAGE ALLOCATIONS IN THE "X" ARRAY  
 X(N1) TO X(N2-1) NC WORDS--STORAGE OF THE CENTR ARRAY  
 X(N2) TO X(N3-1) NC WORDS--STORAGE OF THE NUMBER ARRAY  
 X(N3) TO X(N4-1) NF WORDS--STORAGE OF THE MEMBER ARRAY  
 X(N4) TO X(N5-1) NCWORD WORDS--STORAGE OF THE TOTAL ARRAY  
 X(N5) TO X(N6-1) NV OR NCWORD WORDS--STORAGE OF THE DATA ARRAY  
 X(N6) TO X(N7) NE WORDS--STORAGE OF THE LIST ARRAY IN "RESULT".  
 THE STORAGE OF THE "LIST" ARRAY AFTER THE "DATA" ARRAY NOW IMPLIES THAT  
 "MAX" MUST BE THE VALUE OF "N7".

Figure A.1: Forgy-Jancey Comments

## MODELING OF SHADOWS IN RADAR CLUTTER

```

-----  

*** INPUT SPECIFICATIONS  

CARD 1: TITLE  

CARD 2: PARAMETER CARD  

COLS 1-9 NC NUMBER OF ENTITIES (DATA UNITS)  

COLS 9-13 NC NUMBER OF VARIABLES  

COLS 14-18 NC NUMBER OF CLUSTERS  

COLS 19-20 NC INPUT UNIT FOR THE DATA SET  

NCIN=5, CARD READER  

NCIN=NE, TAKE UP DISK FILE  

COLS 21-22 NTIN=NE,5, DO NOT SAVE CLUSTER MEMBERSHIP LISTS  

NTIN=7, CARD PUNCH  

COLS 23-27 NTOUT=NTOUT UNIT FOR SAVING CLUSTER MEMBERSHIP LISTS  

NTOUT=LP,0, DO NOT SAVE MEMBERSHIP LISTS  

MINREL=TERMINATION PARAMETER, CLUSTERING ENDS WHEN A  

CYCLE THROUGH THE DATA SET RESULTS IN "MINREL"  

NO FURTHER CHANGES IN CLUSTER MEMBERSHIPS  

COLS 28-29 MINREL=L,0, ITERATE TO COMPLETE CONVERGENCE  

IPART=INITIAL PARTITION PARAMETER  

IPART=1, SEED POINTS ARE SELECTED FROM THE DATA UNITS,  

READ THE SEQUENCE NUMBERS FOR THE CHOSEN DATA  

UNITS FROM CARD(S) IN POS16 FORMAT. IF THE  

DATA SET IS NOT STORED IN CORE, THE LIST OF  

SEQUENCE NUMBERS MUST BE IN ASCENDING ORDER  

IPART=2, THE DATA UNITS ARE GROUPED INTO AN INITIAL  

PARTITION IN THE INPUT SEQUENCE WITH THE  

FIRST "NUMBD(1)" IN CLUSTER 1, THE NEXT  

"NUMBD(2)" IN CLUSTER 2 ETC. READ THE  

"NUMBD" ARRAY FROM CARD(S) IN POS16 FORMAT.  

IPART=3, THE SEED VECTORS FOR THE SEED POINTS ARE  

READ FROM CARD(S) & IN FORMAT "FMT" WHICH IS  

READ FROM CARD(S).  

COLS 30-31 METHOD=PARAMETER FOR CHOOSING THE ALGORITHM IN ONE  

VERSITY OF SURROUNING "KMEAN"  

METHOD=1, JANCEY ALGORITHM  

METHOD=4, FORD ALGORITHM  

COLS 32-39 NCMAX=NC NUMBER OF CLUSTERS FOR OUTLIER ROUTINE  

NCMAX=NC, DON'T DO OUTLIER ROUTINE  

NCMAX .LE. 0, USE NC AS NCMAX (I.E. NO LIMIT ON CLUSTERS)  

COLS 40-43 OUTLYR=COEFFICIENT (NPTWFN 3 AND 1 INCLUSIVE) FOR THE  

OUTLIER ROUTINE (NUMBER IN F6.2 FORMAT)  

OUTLYR=1., MAX AMOUNT OF OUTLYING CLUSTERS  

OUTLYR=0., NO OUTLYING CLUSTERS ALLOWED.  

*** CARDS 3 AND 4 ARE READ IN SUBROUTINE "KMEAN" ACCORDING TO THE  

*** PROCEDURE SPECIFIED BY THE CHOSEN VALUE OF "IPART". NOTE THAT THE  

*** BASIC K-MEANS METHOD OF MACQUEEN SIMPLY USES THE FIRST "NC" DATA  

*** UNITS AS CLUSTER SEED POINTS AND THEREFORE IGNORES THE "IPART"  

*** PARAMETER.  

-----  

STORAGE ALLOCATIONS IN THE XNP ARRAY  

X(1) TO X(N2-1) NC*NV WORDS--STORAGE OF THE CENTR ARRAY  

X(N2) TO X(N3-1) NC WORDS--STORAGE OF THE NUMR ARRAY  

X(N3) TO X(N4-1) NC WORDS--STORAGE OF THE MEMR ARRAY  

X(N4) TO X(N5-1) NC*NV WORDS--STORAGE OF THE TOTAL ARRAY  

X(N5) TO X(N6-1) NV OH NC*NV WORDS--STORAGE OF THE DATA ARRAY  

X(N6) TO X(N7) NE WORDS--STORAGE OF THE LIST ARRAY IN "RESULT"  

C THE STORAGE OF THE "LIST" ARRAY AFTER THE "DATA" ARRAY NOW IMPLIES THAT  

-----  

C "MAX" MUST BE THE VALUE OF "N7".  

C*****  

C NOTE:  

C SINCE NC CAN BE INCREASED BY THE OUTLIER ROUTINE, THE  

C ALLOCATIONS IN THE ARRAYS ARE BASED ON NCMAX, NOT NC.  

C THIS CAN INCREASE STORAGE REQUIREMENTS FASTER THAN EXPECTED.  

C*****  


```

**Figure A.2: Outlier Comments**

## MODELING OF SHADOWS IN RADAR CLUTTER

FORTRAN IV G LEVEL 21                    EXEC                    DATE = 79365                    09/26/80

```

0001      *SUBROUTINE EXEC(X,LIMIT)
C
C THIS SUBROUTINE READS PARAMETERS, COMPUTES STORAGE AND CALL'S MAJOR
C PROGRAM SEGMENTS NEEDED FOR A NON-HIERARCHICAL CLUSTERING JOB USING
C ONE OF THE METHODS PROGRAMMED AS A VERSION OF SUBROUTINE "KMEAN".
C
C MORE COMMENTS AND A FULL EXPLANATION CAN BE FOUND IN FILE "DD.FCOMMENT",
C OR BY OBTAINING THE REPORT THAT WAS A SOURCE OF THIS PROGRAM.
C
0002      DIMENSION X(1),TITLE(20)
0003      READ (5,1000) TITLE
0004      READ (5,1100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD
0005      WRITE (6,2000) TITLE
0006      WRITE (6,2100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD
0007      N1=1
0008      N2=N1+NC*NV
0009      N3=N2+NC
0010      N4=N3*NE
0011      N5=N4+NC*NV
C      *N6* MAY (AND WILL) BE INCREASED IN "KMEAN"
0012      N6=45*NV-1
0013      N7=N4+N6-1
0014      MAX=N6
0015      WRITE (6,2200) MAX,LIMIT
0016      IF (MAX .GT. LIMIT) STOP
0017      CALL KMFAN(X(N1),X(N2),X(N3),X(N4),X(N5),NS,NE,NV,NC,NTIN,MINREL,
* IPART,METHOD,LS(1))
      CALL RESULT(X(N1),X(N2),X(N3),X(N4),TITLE,NE,NV,NC,NTOUT,
* X(N4),X(N5))
      WRITE (6,2300)
0019      1000  FORMAT(20A6)
0020      1100  FORMAT(1S15)
0021      2000  FORMAT(1I,20A6)
0022      2100  FORMAT('ONE =',1S,'; NV =',1S,'; NC =',1S,'; NTIN =',1S,';
* 1 NTOUT =',1S,'; MINREL =',1S,'; IPART =',1S,'; METHOD =',1S)
0023      2200  FORMAT('REQUIRED STORAGE =',1S,' WORDS',';
* ALLOCATED STORAGE =',1S,' WORDS')
0024      END
0025

```

Figure A.3: Subroutine EXEC

## MODELING OF SHADOWS IN RADAR CLUTTER

```

FORTRAN IV G LEVEL 21           MAIN           DATE = 79365      10/02/66
0001   C  ALTERED SUBROUTINE EXEC--CALLS KMEAN, RSULT2
0001   C
0001   C THIS SUBROUTINE READS PARAMETERS, COMPUTES STORAGE AND CALL'S MAJOR
0001   C PROGRAM SEGMENTS NEEDED FOR A NON-HIERARCHICAL CLUSTERING JOB USING
0001   C ONE OF THE METHODS PROGRAMMED AS A VERSION OF SUBROUTINE "KMEAN"
0001   C
0001   C MORE COMMENTS AND A FULL EXPLANATION CAN BE FOUND IN FILE "DD.FCOMMENT".
0001   C OR BY OBTAINING THE REPORT THAT WAS A SOURCE OF THIS PROGRAM.
0001   C ---CHANGES MADE TO "N6", "N7", SUBROUTINE "RSULT" IS "RSULT2",
0001   C ---CHANGED INPUT FORMATS (DOCUMENTED IN RSCDE FILE "FCOMMENT")
0002   C
0002   DIMENSION X(1),TITLE(20)
0003   READ (5,1000) TITLE
0004   READ (5,1100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD
0005   WRITE (6,2000) TITLE
0006   WRITE (6,2100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD
0007   N1=1
0008   N2=N1+NC+NV
0009   N3=N2+NC
0010   N4=N3+NE
0011   N5=N4+NC+NV
0012   C  "N6" IS NOT THE SAME AS IN ORIGINAL PROGRAM. ALSO "N7" CHANGED, TOO
0013   N6=N5+NV+NE
0014   N7=N6+NE-1
0015   MAX=N7
0016   IF (MAX .GT. LIMIT) STOP
0017   CALL KMEAN(X(N1),X(N2),X(N3),X(N4),X(N5),NS,NE,NV,NC,NTIN,MINREL,
0018   * IPART,METHOD,1417)
0019   CALL RSULT2(X(N1),X(N2),X(N3),X(N6),TITLE,NE,NV,NC,NTOUT,
0020   * X(N4),X(N5))
0021   RETURN
0022   1000 FORMAT(2I4)
0023   1100 FORMAT(1I2,2I5,2I2,15,2I2)
0024   2000 FORMAT(1I1,25A1)
0025   2100 FORMAT('NE =',19,'/', NV =',19,'/', NC =',19,'/', NTIN =',16,'/
0025   * ', NTOUT =',15,'/, MINREL =',14,'/, IPART =',15,'/, METHOD =',14)
0026   2200 FORMAT('REQUIRED STORAGE =',1R15.10,' WORDS',/
0026   * ', ' ALLOCATED STORAGE =',1R15.10,' WORDS')
0027   END

```

Figure A.4: Subroutine EXEC2

AD-A089 702 CLARKSON COLL OF TECHNOLOGY POTSDAM N Y DEPT OF ELEC--ETC F/G 17/9  
MODELING OF SHADOWS IN RADAR CLUTTER.(U)  
JUL 80 B A BLACK, M AROZULLAH, W LADEW

F30602-78-C-0102

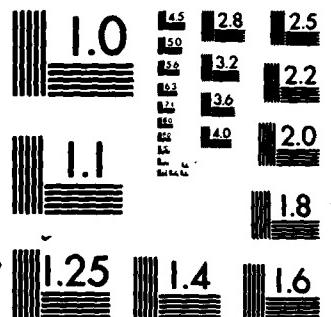
UNCLASSIFIED

RADC-TR-80-238

NL

2<sup>a</sup>  
41  
33

END  
DATE FILMED  
1-150  
DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

# MODELING OF SHADOWS IN RADAR CLUTTER

```

FORTRAN IV G LFVFL 21          MAIN          DATE = 79365        10/21/80
0001      C ALTIMED SUBROUTINE F4CC--CALLS KMEAN?, PSULT2
          C SUBROUTINE F4C(SIX,LM1)
          C
          C THIS SUBROUTINE MEADS PARAMETERS, COMPUTES STORAGE AND CALL'S MAJOR
          C PROGRAM SEPARATES NEEDED FOR A NON-MONOTONIC CLUSTERING AND USING
          C ONE OF THE METHODS DOCUMENTED AS A VERSION OF SUBROUTINE "KMEAN".
          C
          C NOTE: COMMENTS AND A FULL EXPLANATION CAN BE FOUND IN FILE "DDI.FCCMNT"
          C IN BY INITIATING THE REPORT THAT WAS A SOURCE OF THIS PINGPAS.
          C --CHANGES MADE TO "NMCX", SUBROUTINE "PSULT2" IS "PSULT2".
          C --CHANGED INPUT FORMATS (DOCUMENTED IN RSCDC FILE "PCOMM-F77")
          C
          C-----INPUT SPECIFICATIONS-----
          C CARD 1   TITLE
          C CARD 2   PARAMETER CARD
          C          COLS 1-9   NC=NUMBER OF ENTITIES (DATA UNITS)
          C          COLS 9-11  NV=NUMBER OF VARIABLES
          C          COLS 14-18  NC=NUNBER OF CLUSTERS
          C          COLS 19-29  NT (IN=INPUT UNIT FOR THE DATA SET
          C                      NTIN) & CARD HEADER
          C          NTIN=NF=9, TAKE GR DISK FILE
          C          NTOUT=OUTPUT UNIT FOR SAVING CLUSTER MEMBERSHIP LISTS
          C          NTOUTL=9, DO NOT SAVE MEMBERSHIP LISTS
          C          MINREL=TERMINATION PARAMETER, CLUSTERING ENDS WHEN A
          C          CYCLE THROUGH THE DATA SET RESULTS IN "MINREL"
          C          OR FEWER CHANGES IN CLUSTER MEMBERSHIPS
          C          MINREL=.LE.0, ITERATE TO COMPLETE CONVERGENCE
          C          IPART=INITIAL PARTITION PARAMETER
          C          (IPART=1, SEED POINTS ARE SPATIALLY FROM THE DATA UNITS,
          C          READ THE SEQUENC NUMBERS FOR THE CHOSEN DATA
          C          UNITS FROM CARD(S) 3 IN 2016 FORMAT, IF THE
          C-----0001----- KEY0321 COMMENTS DELETED
          C-----0002----- DIMENSION X(1),TITLE(20)
          C-----0003----- READ (5,1001) TITLE
          C-----0004----- 1001 FORMAT (5,1001) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD,NMAX,
          C-----0005----- * OUTLYR
          C-----0006----- WRITE (6,2001) TITLE
          C-----0007----- WRITE (6,2100) NE,NV,NC,NTIN,NTOUT,MINREL,IPART,METHOD,NMAX,
          C-----0008----- * OUTLYR
          C-----0009----- IF (NMAX .LE. 0) NMAX=ONE
          C-----0010----- N1=1
          C-----0011----- N2=N1+NMAX-1
          C-----0012----- N3=N2+NMAX
          C-----0013----- NS=N0+NE+NMAX
          C-----0014----- N7=NS-NE-1
          C-----0015----- NMAX=N7
          C-----0016----- WRITE (6,2200) MAX,LIMIT
          C-----0017----- IF (MAX .GT. LIMIT) CALL SVC(15)
          C-----0018----- CALL KMEAN2(X(N1),X(N2),X(N3),X(N4),X(NS),NS,NE,NV,NC,NTIN,MINREL,

```

```

FORTRAN IV G LEVEL 21          EXEC3          DATE = 79365      10/21/20
                                * IPART,METHOD,LIMIT,NCHMAX,OUTLVR),
0019      * CALL RSLT2(X(N1),X(N2),X(N3),X(N4),TITLE,HE,NV,NC,NTOUT,
                                * X(N5)),
0020      * RETURN
0021      1000  FORMAT(23A4)
0022      1100  FORMAT(15.212,212.15,212,18,F4.2)
0023      2000  FORMAT(1M1,23A4)
0024      2100  FORMAT('ONE',15.,' NV',15.,' NC',15.,' NTIN',15.,
                                * ' NTOUT',15.,' MINDL',15.,' IPART',15.,' METHOD',15.,
                                * ',', ' NCHMAX',15.,' OUTLVR',15.,F4.2)
0025      2200  FORMAT('DREQUIRED STORAGE',15.,' WORDS',15.,
                                * ' ALLOCATED STORAGE',15.,' WORDS',15.)
0026      END

```

**Figure A.5:** Subroutine EXEC3

## MODELING OF SHADOWS IN RADAR CLUTTER

```

FORTRAN IV G LEVEL 21          KNEAN          DATE = 79365      09/5/730
0001    SUBROUTINE KNEAN(CENTR,NUMBR,MEANR,TOTAL,DATA,N5,NC,NV,NC,NTIM,
+  KINNELL,IPART,INITIAL,LIMIT)
C
C-- VERSION 1.0, THE DATA SET IS STORED IN CENTRAL MEMORY.
C
C THIS SUBROUTINE ITERATIVELY SORTS "N5" DATA UNITS INTO "NC" CLUSTERS
C USING THE ALGORITHM OF METMID (NE, 11)
C
C PFGY, F.V., CLUSTER ANALYSIS OF MULTIVARIATE DATA, EFFICIENCY
C VERSUS DETERMINABILITY OF CLASSIFICATION, PAPER PRESENTED AT THE
C BIOMETRIC SOCIETY (UNAMI MEETINGS, RIVERSIDE, CALIFORNIA, JUNE
C 1969). ABSTRACT IN BIOMETRICS, VOLUME 21, NUMBER 3, P 768.
C OR THE ALGORITHM OF (METHOD = 11)
C
C JAMCFY, R.C., MULTIDIMENSIONAL GROUP ANALYSIS, AUSTRALIAN JOURNAL
C OF BOTANY, VOLUME 14, NUMBER 3, APRIL 1966, PP 127-130.
C
C CENTR(1:NV,J-1)=SCORE ON THE I-TH VARIABLE FOR J-TH CLUSTER CENTROID
C TOTAL(ENV(J-1)+1)=TOTAL SCORE ON I-TH VARIABLE FOR DATA UNITS THUS
C FAR ALLOCATED TO THE J-TH CLUSTER
C NUMBR(J)=NUMBER OF DATA UNITS SO FAR ALLOCATED TO THE J-TH CLUSTER
C MEANR(K)=CLUSTER IN WHICH THE K-TH DATA UNIT CURRENTLY BELONGS
C DATA(NV*(K-1)+I)=SCORE ON I-TH VARIABLE FOR K-TH DATA UNIT
C
0002    DIMENSION CENTR(1:NV),TOTAL(1:NV),NUMBR(1:NV),MEANR(1:NV),FMT(20),
+  NAME(6)
0003    DATA NAME/'   F0.0ORGY', ' JA', ' INCEV'/
0004    I=1
0005    IF (METMID .EQ. 11) I=3
0006    WRITE(6,2000) NAME(1),NAME(I+1)
C  CHECK FOR SUFFICIENT STORAGE
0007    N6=N5NF*NV-1
0008    WRITE(6,2100) N6, LIMIT
0009    IF (N6 .GT. LIMIT) STOP
C  ESTABLISH INITIAL PARTITION
0010    IF ((IPART .NE. 3) GO TO 20
C  SEED POINTS ARE READ DIRECTLY FROM CARDS
0011    READ (5,1000) FMT
0012    WRITE(6,2200) FMT
0013    WRITE(6,2300)
0014    JI=0
0015    DO 10 J=1,NC
0016    PEADIS,FMT) (CENTR(J1+I),I=1,NV)
0017    WRITE(6,2400) (CENTR(J1+I),I=1,NV)
10     J1=J1+NV
0018    GO TO 30
C  IPART=1 OR 2
0019    20    WRITE(6,2520) (PART
0020    READ(5,1100) (NUMBR(j),J=1,NC)
0021    WRITE(6,2600) (NUMBR(j),J=1,NC)
0022

```

Figure A.6 (Part 1): Subroutine KNEAN

# MODELING OF SHADOWS IN RADAR CLUTTER

```

FORTRAN IV G LEVEL 21          KMEAN          DATE = 79368        09/31/30
                                C READ THE DATA SET INTO CENTRAL MEMORY
0023      50      K1=1
0024      DD 50 K=1,NK
0025      CALL USER (DATA(K1))
0026      40      K1=K+NV
0027      IF (IPART .EQ. 3) GO TO 100
C     IF IPART IS 1 OR 2 SET UP THE SEED POINTS
0028      IF ((IPART .EQ. 2) .OR. 1) GO TO 60
C     SORT=1. THE DATA UNIT WITH SEQUENCE NUMBER "NUMBER(J)" IS USED AS
C     THE J-TH SPEC POINT
0029      DD 50 J=1,NC
0030      NJ=NUMBER(J)-1+NV
0031      J1=J-1+NV
0032      DD 50 I=1,NV
0033      CFNTK(J1+I)=DATA(NJ+I)
0034      COUNT(J1)=1
0035      GOTO 190
C     IPART=2. THE DATA UNITS ARE GROUPED INTO CLUSTERS WITH THE J-TH
C     CLUSTER HAVING "NJMBR(J)" MEMBERS.
0036      60      K=0
0037      J1=NV
C     ACCUMULATE THE TOTAL SCORE ON EACH VARIABLE FOR EACH CLUSTER
0038      DO 80 J=1,NC
0039      NJ=NUMBER(J)
0040      J1=J1+NV
0041      DD 70 I=1,NV
0042      TOTAL(J1+I)=0
0043      DO 80 K=1,NJ
0044      K=K+1
0045      K1=NMBR(K)+J
0046      K1=(K-1)+NV
0047      DO 80 I=1,NV
0048      J2=J1+I
0049      TOTAL(J2)=TOTAL(J2)+DATA(K1+I)
0050      COUNT(J1)=1
C     COMPUTE THE CENTROIDS
0051      J1=0
0052      DO 90 J=1,NC
0053      DO 90 I=1,NV
0054      J1=J1+1
0055      CFNTK(J1)=TOTAL(J1)/NUMBER(J)
0056      COUNT(J1)=1
0057      G0      COUNT(J1)=1
G1  TO 115
C     INITIALIZE ARRAYS
0058      100     J1=110 K=1,NK
0059      NMBR(K)=0
0060      110     NVAGS=1
C     BEGINNING OF MAIN LOOP
0061      120     J1=0
0062      DO 130 J=1,NC
0063      NUMBER(J)=0
0064      DO 130 I=1,NV
0065      J1=J1+1

```

**Figure A.6 (Part 2): Subroutine KMEAN**

## MODELING OF SHADOWS IN RADAR CLUTTER

```

FORTRAN IV G LEVEL 21          KNEAN          DATE = 79368      09/51/30
0046      130 TOTAL(J1)=0.
0047      MINV=0.0
0048      TDIST=0
0049      C ALLOCATE EACH DATA UNIT TO THE NEAREST CLUSTER CENTROID
0050      K1=0
0051      DD 160 J1=1,NF
0052      KP=101
0053      JP=1
0054      C COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
0055      DNEF=DISTDATA(KP),CFNTRD(J2)
0056      JREFP=1
0057      C TEST DISTANCES TO REMAINING CLUSTER CENTROIDS
0058      DD 140 J2=1,NC
0059      J2=J2+1,NV
0060      DTEST=DISTDATA(KR),CFNTRD(J2)
0061      IF (DTEST .GT. DNEF) GO TO 140
0062      DREF=DTEST
0063      JREF=J
0064      140 CONTINUE
0065      C ALLOCATE DATA UNIT "K1" TO CLUSTER "JREF"
0066      MINREL(JREFP)=NUMREL(JREFP)+1
0067      TDIST=TDIST+JREFP
0068      IF (JREF .EQ. QFNR(.EQ.)) GO TO 150
0069      C THE DATA UNIT CHANGES ITS MEMBERSHIP
0070      NMOVED=NMOVED+1
0071      NF=NR(K),JREF
0072      150 JI=JREF-1,NV
0073      DD 160 I=1,NV
0074      JI=JI+1
0075      KICK=1
0076      TOTAL(JI)=TOTAL(JI)+DATA(KI)
0077      160 CONTINUE
0078      C ALL DATA UNITS ALLOCATED. TEST FOR CONVERGENCE
0079      WRITE(6,779) NMOVED,NPASS,TDIST
0080      NPASS=NPASS+1
0081      JREF=0
0082      IF (.NOT.NMOVED .GT. MINREL) GO TO 165
0083      IF (.NOT.NHMD .NE. 1 .AND. NMOVED .EQ. 0) RETURN
0084      JREFP=1
0085      C COMPUTE TRUE CLUSTER CENTROIDS--FORGY UPDATE
0086      170 JI=0
0087      DD 160 JI=1,NC
0088      DD 160 I=1,NV
0089      JI=JI+1
0090      180 CFNTRD(JI)=TOTAL(JI)/NUMREL(JI)
0091      IF (JREF .NE. 0) RETURN
0092      GO TO 120
0093      C IF (.NOT.NHMD .NE. 1) GO TO 170
0094      C JANKEY UPDATE
0095      190 JI=0
0096      DD 200 JI=1,NC
0097      DD 200 I=1,NV

```

```

FORTRAN IV G LEVEL 21          KMEAN          DATE = 79369      09/31/30
2110      J1=J1+1
0111      CENTR(J1)=2.0*TOTAL(J1)/NUMBR(J1)-CENTR(J1)
0112      GO TO 120
0113      1000  FORMAT(22A6)
0114      1100  FORMAT(20I4)
0115      2000  FORMAT(140.246,
0116      2100  *      METHOD OF CLUSTER ANALYSIS. DATA SET STORED IN CORE*)
      *      (REQUIRED STORAGE =*,15, ' WORDS')
      *      (ALLOCATED STORAGE =*,15, ' WORDS')
0117      2200  FORMAT('FORUMAT',20A6)
0118      2300  FORMAT('!INITIAL CLUSTER CENTERS READ IN AS FOLLOWS',//)
0119      2400  FORMAT(1X,10F12.4)
0120      2500  FORMAT('! I PART =*,12,.', NUMBR ARRAY READ AS FOLLOWS',//)
0121      2620  FORMAT(1X,10I7)
0122      2700  FORMAT('!-0.15,', DATA UNITS MOVED ON ITERATION NUMBER',13,/.
0123      *      ' SUMMED DEVIATIONS ABOUT SEED POINTS =',E16.2)
      END

```

**Figure A.6 (Part 3): Subroutine KMEAN**

## MODELING OF SHADOWS IN RADAR CLUTTER

```

PINTRAN IV G LFVPL PI           KUF4N2          DATE = 79203      09/30/81

0001
C   SUMMITS THE NUMBER OF DATA UNITS, NUMBER, MEMBER, TOTAL, DATA, INDEX, NV, NP, NTIM,
C   * MEMBER(J), PART, GETNM, L1, L1T, NCMAX, QFLYRS.
C
C   SFNPENHOF(J-1)+1=SCORE ON THE I-TH VARIABLE FOR J-TH CLUSTER CENTERED
C   TOTAL(NV) I-1+1=TOTAL SCORE ON I-TH VARIABLE FOR DATA UNITS THIS
C   FAR ALLOCATED TO THE J-TH CLUSTER
C   NMJNBK(J)=NUMBER OF DATA UNITS THIS FAR ALLOCATED TO THE J-TH CLUSTER
C   NMJNBK(J)=NUMBER OF DATA UNITS THIS FAR ALLOCATED TO THE J-TH CLUSTER
C   DATA(INDX(K-1)+1)=SCORE IN I-TH VARIABLE FOR K-TH DATA UNIT
C
C   DIMENSION CENTR(1),TOTAL(1),NUMBER(1),MEMBER(1),DATA(1),PMF(20).
C   * NAME(6)
C   DATA NAME/ 'PINTRAN', 'G', 'LFVPL', 'I', 'JUF4N2', 'SFNPENHOF'
C   101
C   IF (GETNM .EQ. 1) 103
C   UPDT(16,1000) NAME(1),NAME(101)
C
C   ESTABLISH INITIAL PARTITION
C   102  IF ((IPART .EQ. 31) OR (70
C   SEED POINTS ARE READ DIRECTLY FROM CARDS
C   103  READ (5,1001) CNT
C   UPTR(16,2001) CNT
C   WRITE(16,2001) CNT
C   J1=9
C   DD 10 J1,NC
C   PMF(1,PMF) (CENTR(J1+1),1+1,NV)
C   UPTR(16,2001) (CENTR(J1+1),1+1,NV)
C
C   J1=J1+NV
C   10  J1 TO 30
C   101  IPART=1 OR ?
C   20  UPTR(16,1000) (PART
C   PCAD(5,1003) (NMJNBK(J),J=1,NC)
C   UPTR(16,1000) (NMJNBK(J),J=1,NC)
C
C   READ THE DATA SET INTO CENTRAL MEMORY
C   30  K1=1
C   DO 40 K1,M2
C   CALL USER (DATA(K1))
C   40  K1=K1+NV
C
C   IF "IPART" IS 1 OR 2 SET UP THE SEED POINTS
C   50  IF ((IPART .EQ. 31) OR (70 TO 100
C   50  IF ((IPART .EQ. 20, 21) OR (70 TO 60
C   50  IPART=. The data unit with sequence number "NMJNBK(J)" is used as
C   THE J-TH SEED POINT
C
C   DO 50 J=1,NC
C   NJ=(NMJNBK(J)-1)+NV
C   J1=J-1+NV
C   DO 50 J=1,1+NV
C   CENTR(J1+1)=DATA(NJ,0)
C
C   CONTINUE
C   50  GOTO 100
C   IPART=. The data units are grouped into clusters with the J-th
C   cluster having "NMJNBK(J)" members.
C   60  K=0
C   J1=1+NV

```

**Figure A.7 (Part 1): Subroutine KMEAN2**

## MODELING OF SHADOWS IN RADAR CLUTTER

```

PORTMAN IV 6 LEVEL 21          KMEANS          DATE = 79368        09/26/81
                                TOTAL = 79368

0038      C ACCUMULATE THE TOTAL SCORE ON EACH VARIABLE FOR EACH CLUSTER
0039      DO 70 K=1,NCL
0040      KJ=NK(K)
0041      J1=1,NV
0042      DO 70 J=1,NV
0043      KJK=NK(J)
0044      KJ=NK-1,NV
0045      DO 70 J=1,NV
0046      JK=NK(J)
0047      TOTAL(JK)=TOTAL(JK)+DATA(K,J)
0048      CONTINUE
0049      C COMPUTE THE CENTROIDS
0050      J1=0
0051      DO 90 J=1,NC
0052      DO 90 J=1,NV
0053      J1=J1+1
0054      C TEST FOR NUMRA(J)=TOTAL(J1)/NUMR(J)
0055      CNTA(J1)=TOTAL(J1)/NUMR(J)
0056      90  CONTINUE
0057      GO TO 115
0058      C INITIALIZE ARRAYS
0059      I03  DO 110 K=1,NC
0060      I10  NEMR(K)=0
0061      I15  NDASG=0
0062      C BEGINNING OF MAIN LOOP
0063      120  J1=0
0064      DO 130 J=1,NMAX
0065      NUMR(J)=0
0066      DO 130 I=1,NV
0067      J1=J1+1
0068      TOTAL(J1)=0.
0069      MOVE3=0
0070      TDI5T=0
0071      C ALLOCATE EACH DATA UNIT TO THE NEAREST CLUSTER CENTROID
0072      K1=0
0073      DO 160 K=1,NC
0074      KPK=1+1
0075      J2=1
0076      C COMPUTE DISTANCE TO FIRST CLUSTER CENTROID
0077      DREF=DIST(DATA(KP),CENTA(J2))
0078      SUMDIST=DREF
0079      JREF=1
0080      C TEST DISTANCES TO REMAINING CLUSTER CENTROIDS
0081      DO 160 J=2,NV
0082      JPA=J2+NV
0083      DIST=DATA(KP),CENTR(JPA)
0084      SUMDIST=SUMDIST+DIST
0085      IF (SUMDIST.GT.0.0) DREF=SUMDIST/DO 160
0086      DREF=DREF/2
0087      JREF=J

```

**Figure A.7 (Part 2): Subroutine KMEAN2**

## MODELING OF SHADOWS IN RADAR CLUTTER

FORTRAN IV G LEVEL 21 KMEAN2 DATE = 79365 09/26/81  
 0000 140 CONTINUE  
 0001 140 CONTINUE,NM=1,NC  
 C TEST FOR CONVERGENCE, AND ALLOCATE NEW CLUSTERS IF ALLOWED.  
 0002 IF (LWIL, ITAII, LWMR>0) GO TO 145  
 \* AND, INC(LT, NCWAKI)) GO TO 145  
 C MEET NEW CLUSTER  
 0003 NCWAKI  
 0004 JREF=NC  
 0005 DO 145 I=1,NC  
 C PNTPHV(JREF-1)OL)=DATA(IK1D)  
 0006 145 CONTINUE  
 C ALLOCATE DATA UNIT #LT TO CLUSTER "JREF"  
 0007 NMNBR(JREF)=NMNBR(JREF)+1  
 0008 10151121567161  
 0009 IF (JREF .EQ. 0, NMNBR(I)) GO TO 180  
 C THE DATA UNIT CHANGES ITS MEMBERSHIP  
 0010 NMV,SMOVES,I  
 0011 4F 010121567161  
 0012 150 JI=(JREF-1)\*NV  
 0013 DO 160 I=1,NV  
 0014 JI=JI+1  
 KIK1D  
 TOTAL(JI)=TOTAL(JI)+DATA(IK1D)  
 160 CONTINUE  
 C ALL DATA UNITS ALLOCATED, TEST FOR CONVERGENCE  
 0009 WRITE(6,\*700) NMV,NPASS,TOLST,NC  
 NMASS=NPASS+1  
 0100 JREF=0  
 0101 IF (NMVES .GT. NMNL) GT TO 185  
 0102 IF (METHOD .NE. 1 .AND. NMVES .EQ. 0) RETURN  
 0103 JREF=1  
 C COMPUTE TRUE CLUSTER CENTROIDS--FORGY UPDATE  
 0104 170 JI=0  
 0105 DO 180 J=1,NC  
 0106 DO 180 I=1,NV  
 0107 JI=JI+1  
 0108 IF ((NMNR(J) .LE. 0) GO TO 180  
 0109 CENTR(JI)=TOTAL\_(JI)/NMNR(J)  
 0110 180 CONTINUE  
 0111 IF (JREF .EQ. 1) RETURN  
 0112 DO 125  
 0113 C  
 0114 185 IF (METHOD .NE. 1) GO TO 170  
 C JANCO UPDATE  
 0115 190 JI=0  
 0116 DO 200 J=1,NC  
 0117 DO 200 I=1,NV  
 0118 JI=JI+1  
 C TEST FOR NMNR=0 HERE, TOO...  
 200 CE4TR(JI)=2.0\*TOTAL(JI)/NMNR(J)-CENTR(JI)  
 0119 200 GO TO 120  
 0120 1030 FORMAT(2D4)  
 0121 1100 FORMAT(2D4)  
 0122

FORTRAN IV G LEVEL 21 KMEAN2 DATE = 79365 09/26/81  
 0123 2000 FORMAT(1H0,24A,  
 \* , METHOD OF CLUSTER ANALYSIS. DATA SET STORED IN CORE')  
 0124 2100 FORMAT(' REQUIRED STORAGE ',15,' WORDS',/)  
 \* ' ALLOCATED STORAGE ',15,' WORDS')  
 0125 2200 FORMAT('FORMAT',20A1)  
 0126 P355 FORMAT('INITIAL CLUSTER CENTERS READ IN AS FOLLOWS',//)  
 0127 2300 FORMAT(1X,1D12.6)  
 0128 2500 FORMAT(1X,1D12.6, NMNR ARRAY READ AS FOLLOWS',//)  
 0129 2600 FORMAT(1X,1D17)  
 0130 2700 FORMAT(1H0,15,' DATA UNITS MOVED ON ITERATION NUMBER',15,/  
 \* , SUMMED DEVIATIONS ABOUT SEED POINTS ',E16.2,10E-15, ' NC ',15)  
 END

Figure A.7 (Part 3): Subroutine KMEAN2

## MODELING OF SHADOWS IN RADAR CLUTTER

FORTRAN IV G LEVEL 21                    RESULT                    DATE = 79365                    10/29/82

```

0001         SUBROUTINE RESULT(CENTR,NV,NV,LIST,TITLE,VI,MV,NC,NINTUT,
0001                  & TOTAL,DATA)
0002         C THIS SUBROUTINE INPUTS THE RESULTS FROM A CLUSTERING JOB BASED
0002                  ON ANY VERSION OF SUBROUTINE INC. "MEAN".
0003         DIMENSION CENTR(1),NV(1),LIST(1),TITLE(20),TOTAL(1),
0003                  DATA(1)
0004         C AS A CONTINUATION PROCEDURE WRITE OUT THE RAW MEMBERSHIP LIST.
0004         WRITE(6,2000) TITLE
0005         WRITE(6,2100) (NV(K),K=1,NC)
0006         WRITE(6,2200) (NINTUT(J),J=1,NC)
0007         C INITIATE THE "MEMBR" ARRAY AND PUT THE RESULT IN THE "LIST" ARRAY.
0008         C FURTHER REVERSE THE "NUMBR" ARRAY TO CONTAIN START POINTS IN THE
0009                  ALLOCATED ARRAY FOR EACH CLUSTER
0010         NUMBR(NC)=NE-NUMBR(NC)+1
0011         J=NC
0012         DO 10 J=J-1,NC
0013         NV(J)=NUMBR(J+1)-NUMBR(J)
0014         J=J-1
0015         CONTINUE
0016         C BUILD "LIST" ARRAY
0017         DO 20 K=1,NF
0018         IF(MARK(K).NE.WOR(K))
0019                  NV(K)=NUMBR(NMARK)
0020         NV(K)=NUMBR(NMARK)+1
0021         LIST(NV)=K
0022         NV=NMARK+1
0023         NV=NUMBR(NMARK)+1
0024         NV=NV+1
0025         NV=NV+1
0026         NV=NV+1
0027         NV=NV+1
0028         NV=NV+1
0029         C PRINT RESULTS FOR EACH CLUSTER
0030         WRITE(6,2000) TITLE
0031         K=1
0032         DO 30 J=1,NC
0033         WRITE(6,2300) J,NV(J)
0034         JI=(J-1)*NV
0035         WRITE(6,2400) (CENTR(JI+I),I=1,MV)
0036         K2=K1+NV(J)-1
0037         WRITE(6,2500) (LIST(K),K=K1,K2)
0038         K1=K2+1
0039         CONTINUE
0040         C THIS DUMPS OUT "TOTAL" ARRAY

```

FORTRAN IV G LEVEL 21                    RESULT                    DATE = 79365                    10/29/82

```

0041         DD 62 (71,NC
0042         4000 FORMAT('TOTALS FOR CLUSTER',I3)
0043         4100 FORMAT(IX,E10.5)
0044         60 CONTINUE
0045         RETURN
0046         2000 FORMAT(1M1,20A6)
0047         2100 FORMAT('DRAW MEMBERSHIP LIST',/(1X,25IS))
0048         2200 FORMAT('CLUSTER SIZES',/(1X,25IS))
0049         2300 FORMAT('CLUSTER',I3,' CONTAINS',15,' DATA UNITS')
0050         2400 FORMAT('CENTR',I3,' COORDINATES',/(1X,10E12.4))
0051         2500 FORMAT('MEMBERSHIP LIST',/(1X,25IS))
0052         3000 FORMAT(25A6)
0053         3100 FORMAT(16IS)
0054         END

```

Figure A.8: Subroutine RESULT

## MODELING OF SHADOWS IN RADAR CLUTTER

```

UNITN IV S LEVEL 21          RESULT#          DATE + 70308        10/31/13
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
      SUBROUTINE NSU_T9(CENTR,NUMIN,NUMINL,LIST,TIT,T,NV,NC,OUTUT,
      TOTAL,DATA)
C THIS SUBROUTINE WRITES THE RESULTS FROM A CLUSTERING JOB BASED
C ON ANY VERSION OF SIMULUS FOR "KUEAN".
C
C   DIMENSION CENTR(1),NUMIN(1),NUMINL(1),LIST(1),TITLE(20),TOTAL(1),
C   * DATA(1)
C AS A CONTINUATION OF CAUTION WRITE OUT THE RAW MEMBERSHIP LIST.
      WRITE(6,2000) TITLE
      WRITE(6,2100) (NUMBER(K),K=1,NC)
      WRITE(6,2200) (NUMBER(J),J=1,NV)
C INPUT THE "LIST" ARRAY AND PUT THE RESULT IN THE "LIST" ARRAY.
C FIRST REVERSE THE "NUMIN" ARRAY TO CONTAIN START POINTS IN THE
C "LIST" ARRAY FOR EACH CLUSTER
      NUMBER(NC)=NE-NUMBR(NC)+1
      J=NC
      J=J-1
      DO 10 J=2,NC
      NUMBER(J)=NUMBER(JJ)-NUMBER(JJ+1)
      J=J+1
      J=J-1
      10 CONTINUE
C BUILD "LIST" ARRAY
      DO 20 K=1,NC
      NUMBER(K)=NUMBER(K)
      NJ=NUMBER('CNHAC')
      LIST(NJ)=K
      NUMBER(4*NC)=NUMBER(MEMBER)+1
      20 CONTINUE
C SAVE THE SORTED MEMBERSHIP LIST IF DESIRED
      IF (INTOUT .EQ. 2) GO TO 30
      WRITE (INTOUT,3001) TITLE
      WRITE (INTOUT,3100) (LIST(K),K=1,NC)
      DO 22 K=1,NC
      22 CALL USRROUT(DATA,(LIST(K)-1)*NV+1))
C RESTORE THE "NUMBER" ARRAY.
      30 J=NC
      DO 40 J=2,NC
      NUMBER(J)=NUMBER(JJ)-NUMBER(JJ-1)
      J=J-1
      40 CONTINUE
C PRINT RESULTS FOR EACH CLUSTER
      WRITE(6,2000) TITLE
      K1=1
      DO 50 J=1,NC
      WRITE(6,2300) J,NUMBER(J)
      J1=(J-1)*NV
      JNTE=(J-1)*NV
      K2=K1+NUMBER(J)-1
      WRITE(6,2400) (CENTR(J1+I),I=1,NV)
      K2=K1+NUMBER(J)-1
      WRITE(6,2500) (LIST(K),K=K1,K2)
      K1=K2+1
      50 CONTINUE

```

```

FORTRAN IV 6 LEVEL 21          RESULTS          DATE = 79365      10/31/13

0040      20  CONTINUE
0041      C THIS DUMPS OUT "TOTAL" ARRAY
0042      DO 60 J=1,NVC
0043      WRITE(6,100) TOTAL(J)
0044      1000  FORMAT('TOTALS FOR CLUSTER',I3)
0045      6100  FORMAT(1X,E10.5)
0046      60  CONTINUE
0047      RETURN
0048      2000  FORMAT(1X,20A6)
0049      2100  FORMAT('ORAN MEMBERSHIP LIST',/,1X,25(1))
0050      2200  FORMAT('CLUSTER SIZES',/,1X,25(1))
0051      2300  FORMAT('CLUSTER',1X,'CONTAINS',1X,' DATA UNITS')
0052      2400  FORMAT('OCENTERID',1X,'COORDINATES',/,1X,100(2.0))
0053      2500  FORMAT('MEMBERSHIP LIST',/,1X,25(1))
0054      3000  FORMAT(23A6)
0055      3100  FORMAT(10E13)
0056      END

```

**Figure A.9:** Subroutine RSULT2

## MODELING OF SHADOWS IN RADAR CLUTTER

```

FORTRAN IV G LEVEL 21          MAIN           DATE = 79369    11/19/01
                                C THIS IS THE DRIVING PROGRAM. FURTHER DOCUMENTATION IN ROSCOE FILE "FCOM"
0001      DIMENSION X(4000)
0002      LIMIT=20000
0003      CALL EXEC PIX,LIMIT)
0004      END FILE 2
0005      STOP
0006      PNO

```

```

FORTRAN IV G LEVEL 21          DIST           DATE = 79369    11/19/01
                                FUNCTION DIST(X,Y)
0001      DIMENSION X(1),Y(1)
0002      DIST=0.
0003      DC 10 [1] .2
0004      10  DIST=DIST+(X(1)-Y(1))**2
0005      RETURN
0006      END

```

```

FORTRAN IV G LEVEL 21          USER           DATE = 79369    11/19/01
                                SUBROUTINE USER(Z)
0001      DIMENSION X(1)
0002      READ(5,100)(X(I),I=1,3)
0003      100  FORMAT(2F4.0,E17.10)
0004      RETURN
0005      END

```

```

FORTRAN IV G LEVEL 21          USROUT         DATE = 79369    11/19/01
                                SUBROUTINE USROUT(X)
0001      DIMENSION X(1)
0002      WRITE(2,10)(X(I),I=1,3)
0003      100  FORMAT(2F4.0,E17.10)
0004      RETURN
0005      END

```

**Figure A.10: MAIN; USER; USROUT; Function DIST**

## MODELING OF SHADOWS IN RADAR CLUTTER

FORTRAN IV G LEVEL 21 MAIN DATE = 00111 06/59/00

```
C DRAWN. FUNCTION LOC. IN FILE 'COMMENT'  
0001      DIMENSION X(24300)  
0002      LIMIT=24300  
0003      CALL EXEC3(X,LIMIT)  
0004      END FILE 4  
0005      STOP  
0006      END
```

FORTRAN IV G LEVEL 21 DIST DATE = 00111 06/59/00

```
0001      FUNCTION DIST(X,Y)  
0002      DIMENSION X(1),Y(1)  
0003      C DISTANCE IS EUCLIDIAN (X-Y)  
0004      DIST=0.0  
0005      DO 10 I=1,2  
0006      10 DIST=DIST+(X(I)-Y(I))**2  
0007      RETURN  
0008      END
```

FORTRAN IV G LEVEL 21 DIST DATE = 00110 06/02/30

```
0001      FUNCTION DIST(X,Y)  
0002      DIMENSION X(1),Y(1)  
0003      C THIS DISTANCE FUNCTION ONLY USES THE THIRD VARIABLE--INTENSITY  
0004      DIST=AUS(X(3)-Y(3))  
0005      RETURN  
0006      END
```

Figure A.11: MAIN; Two Versions of DIST

## MODELING OF SHADOWS IN RADAR CLUTTER

```
FORTRAN IV G LEVEL 21          RAY          DATE = 79365      10/30/89
0001      SUBROUTINE RAY(IX,IY,YFL,SIGMA)
0002      IX=IX85539
0003      IF ((IY .LE. 0) GOTO 10
0004      IY=IY+214748364761
0005      10      YFL=IY
0006      YFL=YFL*0.4656613E-9
0007      YFL=SQRT((-2.0*SIGMA**2*4LOG(1-YFL)))
0008      IX=IX
0009      RETURN
0010      END
```

Figure A.12: Subroutine RAY

## MODELING OF SHADOWS IN RADAR CLUTTER

FORTRAN IV G LEVEL 21	MAIN	DATE = 79365      10/26/16
0001	DIMENSION VALUE(100,100)	
0002	SIGMA1=1.	
0003	SIGMA2=10000.	
0004	I=X=123456789	
0005	DO 100 I=1,100	
0006	DO 100 J=1,100	
0007	CALL RAY(I,X,IY,YFL,SIGMA1)	
0008	100      VALUE(I,J)=YFL	
0009	DO 1 I=1,10	
0010	DO 1 J=1,100	
0011	CALL RAY(I,X,IY,YFL,SIGMA2)	
0012	1      VALUE(I,J)=VALUE(I,J)+YFL	
0013	DO 2 I=20,30	
0014	DO 2 J=1,30	
0015	CALL RAY(I,X,IY,YFL,SIGMA2)	
0016	2      VALUE(I,J)=VALUE(I,J)+YFL	
0017	DO 3 I=20,30	
0018	DO 3 J=1,100	
0019	CALL RAY(I,X,IY,YFL,SIGMA2)	
0020	3      VALUE(I,J)=VALUE(I,J)+YFL	
0021	DO 4 I=31,50	
0022	DO 4 J=1,100	
0023	CALL RAY(I,X,IY,YFL,SIGMA2)	
0024	4      VALUE(I,J)=VALUE(I,J)+YFL	
0025	DO 5 I=60,80	
0026	DO 5 J=1,50	
0027	CALL RAY(I,X,IY,YFL,SIGMA2)	
0028	5      VALUE(I,J)=VALUE(I,J)+YFL	
0029	DO 6 I=60,80	
0030	DO 6 J=1,100	
0031	CALL RAY(I,X,IY,YFL,SIGMA2)	
0032	6      VALUE(I,J)=VALUE(I,J)+YFL	
0033	DO 7 I=81,100	
0034	DO 7 J=1,100	
0035	CALL RAY(I,X,IY,YFL,SIGMA2)	
0036	7      VALUE(I,J)=VALUE(I,J)+YFL	
0037	DO 300 I=1,100	
0038	DO 300 J=1,100	
0039	300      WRITE(1,500) I,J,VALUE(I,J)	
0040	500      FORMAT(2I4,F17.10)	
0041	END FILE 1	
0042	WRITE(6,1000) IV	
0043	1000      FORMATT(IX,IIII)	
0044	END	

Figure A.13: Creates Rayleigh Data, SIGMA $\pm$ 10,000.

## MODELING OF SHADOWS IN RADAR CLUTTER

```

      DIMENSION(IX,IY) VALUE(100,100)
      SIGNAL1=1.
      SIGNAL2=50.
      IX=92247045
      DO 100 I=1,100
      DO 100 J=1,100
      CALL RAY(I,IX,IY,YFL,SIGNAL1)
      100   VALUE(I,J)=YFL
      DO 1  I=1,9
      DO 1  J=1,100
      CALL RAY(I,IX,IY,YFL,SIGNAL2)
      1     VALUE(I,J)=VALUE(I,J)+YFL
      DO 2  I=2,30
      DO 2  J=1,39
      CALL RAY(I,IX,IY,YFL,SIGNAL2)
      2     VALUE(I,J)=VALUE(I,J)+YFL
      DO 3  I=20,30
      DO 3  J=51,100
      CALL RAY(I,IX,IY,YFL,SIGNAL2)
      3     VALUE(I,J)=VALUE(I,J)+YFL
      DO 4  I=31,59
      DO 4  J=1,100
      CALL RAY(I,IX,IY,YFL,SIGNAL2)
      4     VALUE(I,J)=VALUE(I,J)+YFL
      DO 5  I=62,80
      DO 5  J=1,59
      CALL RAY(I,IX,IY,YFL,SIGNAL2)
      5     VALUE(I,J)=VALUE(I,J)+YFL
      DO 6  I=63,90
      DO 6  J=91,100
      CALL RAY(I,IX,IY,YFL,SIGNAL2)
      6     VALUE(I,J)=VALUE(I,J)+YFL
      DO 7  I=81,100
      DO 7  J=1,100
      CALL RAY(I,IX,IY,YFL,SIGNAL2)
      7     VALUE(I,J)=VALUE(I,J)+YFL
      DO 300 I=1,120
      DO 300 J=1,100
      300   WRITE(11,500) I,J,VALUE(I,J)
      500   FORMAT(2I4,E17.13)
      END FILE 1
      WRITE(6,1000) IX
      1000  FORMAT(IX,1L1)
      END

```

Figure A.14: Creates Rayleigh Data, SIGMA=50.

## MODELING OF SHADOWS IN RADAR CLUTTER

FORTRAN IV G LEVEL 21 PLOT DATE = 79365 10/02/81

```
0191      SUBROUTINE RDPRT(UNIT1,ICOL)
0003      LOGICAL UNIT1,RCRDR,RCRDR5(50),DDT
0004      DATA CHARS/1,2,3,4,5,6,7,8,9,0,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z/
0005      DATA DDT/1.0/
0006      DO 1 J=1,100
0007      1  DOX(I,J)=DDT
0008      DO 10 I=1,100
0009      10  I=I+.9999
0010      READ(15,20,END=15)KOUNT
0011      20  FORMAT(15)
0012      DO 30 NUMBER=1,KOUNT
0013      READ(15,40,END=16) DATA1,DATA2
0014      40  FORMAT(2I10)
0015      INT1=INT(DATA1)
0016      INT2=INT(DATA2)
0017      DOX(INT1,INT2)=CHARS(I)
0018      CONTINUE
0019      30  CONTINUE
0020      WRITE(16,55)
0021      55  FORMAT(' *** ERROR *** END OF DATA SET ON UNIT 01.**')
0022      15  DO 100 I=1,100
0023      100  WRITE(16,200)(DOX(I,J),J=1,100)
0024      200  FORMAT(1X,129I1)
0025      STOP
      END
```

Figure A.15: Cluster Plot Program

## MODELING OF SHADOWS IN RADAR CLUTTER

FORTRAN IV G LEVEL 21	MAIN	DATE = 80316	80/35/06
-----------------------	------	--------------	----------

```

C GRAY SCALE PROGRAM...
0001      INTGEN, GRAY
          READ ALINE(120), THRESH(0)
0002      LOGICAL, SCALE(1), PLINE(120)
0003      C READ IN THE GRAY SCALE SYMBOLS...
0004      READ(SA55)(SCALE(I),I=1,9)
0005      S      FORMAT(9A1)

C READ THE LOWEST VALUE FOR EACH GRAY LEVEL.
0006      DO 10 I=2,8
0007      READ(SA100) THRESH(I)
0008      I=1-1
0009      *PLINE(6,101) SCALE(I),THRESH(I)
0010      10  CONTINUE
0011      *PLINE(6,102) SCALE(3),THRESH(3)
C READ MAX AND MIN. AXIS VALUES. (C= 1281 <= 256)
0012      READ(SA130) MAXV,MINV
0013      WRITE(SA201) MAXV,MINV
0014      DO 20 J=1,MAXH
0015      READ(SA300) XLINE(J)
0016      20  CONTINUE
0017      GRAY=1
0018      DO 40 K=2,8
0019      IF (XLINE(J) .LE. THRESH(K)) GO TO 40
0020      GRAY=K
0021      40  CONTINUE
0022      PLINE(J)=SCALE(GRAY)
0023      50  LUNITNE
0024      *PLINE(6,103)(PLINE(J),J=1,MAXH)
0025      20  CONTINUE
0026      STOP
0027      100   FORMAT(F15.7)
0028      101   FORMAT(' SY430',A1,'<',F15.7)
0029      102   FORMAT(' SY430',A1,'>',F15.7)
0030      201   FORMAT(' MAXV=',I4,' MAXH=',I4,',/,'I4)
0031      200   FORMAT(2I4)
0032      400   FORMAT(' ',128A1)
0033      300   FORMAT(8X,E12.7)
0034      END

```

FIGURE A.16: Grey Scale Program

## MODELING OF SHADOWS IN RADAR CLUTTER

### REFERENCES

1. R.C. Jancey, "Multidimensional Group Analysis," Aust J Bot. vol 14, no 1, April 1966, pp 127-130.
2. C.T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," IEEE Trans on Computers. vol C-20, no 1, January 1971, pp 68-86.
3. Statement of Work for "Topographic Map/Backscatter Correlation," RFP Solicitation Number F30602-79-R-0205, RADC, Griffiss Air Force Base, New York, June 1979, pp 3,4.
4. F.E. Nathanson, Radar Design Principles. New York: McGraw-Hill Book Co, 1969.
5. D.K. Barton, Radars, vol 5: Radar Clutter. Dedham, Mass: Artech House, Inc, 1975.
6. R.M. Cormack, "A Review of Classification," J Royal Stat Soc. series A, vol 134, part 3, 1971, pp 321-367.
7. Documentation for ARTHUR. distributed by A.M. Harper, Laboratory for Chemometrics, Univ of Georgia, Athens, Ga, 30602.
8. B.A. Black and N. Arozullah, "Modeling of Shadowing in Radar Returns," Technical Report prepared under Postdoctoral Program Contract F30602-78-C-0102, Task 5, A-8-1142, RADC, Griffiss Air Force Base, New York, November 1978.
9. W.A. Fordon, "Statistical Pattern Recognition Techniques as Applied to Radar Returns," Technical Report prepared under Postdoctoral Program Contract F30602-78-C-0102, RADC, Griffiss Air Force Base, New York, 1978.
10. E.W. Forgy, "Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications," Biometrics Abstracts. vol 21, no 3, Sept 1965, pp 768-769.

